Departamento de Métodos Cuantitativos y Sistemas de Información
Facultad de Agronomía
Universidad de Buenos Aires

# Introducción a la Programación para las Ciencias Ambientales

## Guía de clases teóricas y prácticas

**CENTRO DE IMPRESIONES**
CIFA 2009

# *Cátedra de Métodos Cuantitativos Aplicados*

***Equipo docente de la cátedra***

Ing. Agr. Susana B. Perelman (Profesora Asociada- Coordinadora de la cátedra)
Ing. Agr. Norberto J. Bartoloni (Profesor Asociado)
Lic. Mirta González (Profesora Asociada)
Ing. Agr. M. V. López (Profesora Adjunta)
Ing. Agr. William B. Batista (Profesor Adjunto)
Lic. Olga S. Filippini (Profesora Adjunta)
Lic. Víctor Brescia (Profesor Adjunto)
Lic. María C. Fabrizio (Jefe de TP)
Ing. Agr. Rosa T. Boca (Jefe de TP)
Ing. Civ. Diana Giorgini (Jefe de TP)
Ing. Agr. Juan P. Guerschman (Jefe de TP)
Ing. Agr. Gustavo A. Sznaider (Jefe de TP)
Dr. Karina Hodara (Jefe de TP)
Ing. Agr. Laura Puhl (Jefe de TP)
Ing. Agr. Pablo A. Cipriotti (Jefe de TP)
Lic. Alexis Cerezo (Ayud. 1era)
Lic. Roxana Aragón (Jefe de TP)
Ing. Agr. Luis Frank (Jefe de TP)
Ing. Agr. Lucas Garibaldi (Ayud. 1era)
Ing. Agr. Gonzalo Grigera (Ayud. 1era)
Lic. Fernando Biganzoli (Ayud. 1era)
Ing. Agr. Cecilia Conde (Ayud. 1era)
Lic. Ezequiel Ramos (Ayud. 1era)
Sebastián Giedzinski (Ayudante alumno)
Erika Hirschowitz (Ayudante alumna)
Ileana Boyle (Ayudante alumna)
Facundo Gallo (Ayudante alumno)

Pabellón Wernicke (Ver plano en www.agro.uba.ar/fauba/ubicacion/plano.htm)
**Teléfono:** 4524-8077 **Secretaria:** Ana Inés Garaño e-mail: garano@agro.uba.ar
Página web cátedra www.agro.uba.ar/catedras/metodos/index.htm
Página web curso www.agro.uba.ar/carreras/ambientales/materias/programacion/index.htm

### *Carreras de Agronomía, Ciencias Ambientales y Gestión de Agroalimentos*

Estadística general (cuatrimestral, 5 hs. semanales, se dicta en ambos cuatrimestres)
Modelos Estadísticos (bimestral, 6hs. semanales, se dicta en 1er y 3er bimestres)
Programación para Ciencias Ambientales (bimestral, 5hs. semanales, 1er bimestre)
*Licenciatura en Economía y Administración Agrarias*
Estadística I (cuatrimestral, 6horas semanales, 2° cuatrimestre)
Econometría (cuatrimestral, 4horas semanales, 1er cuatrimestre)
Estadística para Administradores (cuatrimestral, 3horas semanales, 1er cuatrimestre)

### *Carrera técnica en Floricultura*

*Estadística (bimestral, 2 horas semanales, 2° bimestre)*

### *Cursos de intensificación y especialización*

Estadística aplicada a la protección vegetal
Colección e Interpretación de datos
*Cursos de Maestría y Doctorado en la EPG- FAUBA*
Estadística Aplicada - Maestría en Agronegocios
Estadística Aplicada a la Investigación Biológica - Maestría en Biometría y Mejoramiento
Análisis Multivariado Descriptivo en Investigaciones Biológicas y Sociales- Maestría en Biometría y Mejoramiento
Genética de Poblaciones y Evolución - Maestrías de Producción Vegetal y Rec. Naturales
Genética de Poblaciones - Maestría en Biometría y Mejoramiento
Introducción a la Estadística - Maestría en Economía Agraria
Teoría Estadística - Maestría en Biometría y Mejoramiento
Análisis de la Heterogeneidad de la Vegetación - Maestría en Recursos Naturales
Análisis de Datos Categóricos- Maestría en Biometría y Mejoramiento

# Índice

# Cronograma 1^er^ Bimestre 2009

| Fecha | Clase | Tema |
|---|---|---|
| 10/03 | 1 | Introducción a los modelos de simulación y bases de datos en las ciencias ambientales. *Entrega de consigna "Tablas Dinámicas" y video.* |
| 12/03 | 2 | Definición de base de datos. Dato. Información. Campo. Registro. Gestores de base de datos. Ejemplos. |
| 17/03 | 3 | Creación de una base de datos: Modelo lógico. Modelo físico. Confección de la base de datos en un gestor. *Entrega de consigna de trabajo domiciliario* **(presentación estado de avance clase 7, presentación final clase 14)** |
| 19/03 | 4 | Uso de una base de datos. Ingreso de información. Ordenamiento. Importación y vinculación. Recuperación de datos. Consultas. |
| 24/03 | | **Feriado** |
| 26/03 | 5 | Construcción de modelos. Objetivo, supuestos, límites, diagrama de flujo, sistema simbólico, variables de entrada y salida, algoritmos matemáticos, decisiones condicionales. Presentación *Modelo de almacenaje de agua en el suelo en Excel.* **Entrega consigna Ejercicio Integrador.** |
| 31/03 | 6 | **Ejercicio Integrador** y Programación de modelos. Presentación del ambiente de programación *Stella*: áreas de trabajo, construcción del modelo simbólico y los algoritmos matemáticos. **Entrega de consigna trabajo domiciliario dinámica Carbono (estado de avance clase 12, presentación final clase 15).** |
| 02/04 | | **Feriado** |
| 07/04 | 7 | Modelización en Stella 1. Ecuaciones diferenciales. Algoritmos. Mecanismos de retroalimentación. Desarrollo del *Modelo de almacenaje de agua en el suelo* en Stella. Entrega **Anexo Ecuaciones diferenciales.**<br>**Entrega estado de avance** del trabajo grupal domiciliario Bases de datos |
| 09/04 | | **Feriado** |
| 14/04 | 8 | Modelización en Stella 2. Organización de los resultados. Interfase gráfica. Estudio del comportamiento del sistema mediante ejercicios de simulación. Desarrollo de modelos poblacionales sencillos. Alcances y limitaciones del ambiente de programación. |
| 16/04 | 9 | Evaluación de modelos: Calibración, verificación, validación, análisis de sensibilidad y optimización de los parámetros de modelos. Modelo simulador forestal o almacenaje.<br>Entrega **estado de avance** del trabajo grupal domiciliario. Modelo de dinámica del carbono. |

| 21/04 | 10 | Entrega y exposición trabajo Bases de Datos |
|---|---|---|
| 24/04 | 11 | Entrega y exposición trabajo dinámica carbono. |
| 27/04 | 12 | Examen |
| 30/04 | 13 | |

# Régimen de aprobación de la materia

Examen integrador base de datos: calificación máxima **30 puntos**
Examen integrador: calificación máxima **40 puntos**

Trabajos domiciliarios base de datos: calificación máxima **20 puntos**
Trabajos domiciliarios modelo de simulación: calificación máxima **10 puntos**

- Los alumnos que en su evaluación general acumulen **70** (setenta) o más puntos, aprueben los trabajos domiciliarios y cumplan con la asistencia obligatoria, promocionarán la materia,
- aquellos que acumulen entre **40** (cuarenta) y **69** (sesenta y nueve) puntos, presenten los **trabajos prácticos domiciliarios** en tiempo y forma, y cumplan con la asistencia obligatoria alcanzarán la regularidad y
- quienes acumulen **menos de 40** puntos y/o **no entreguen** los trabajos domiciliarios y/o **no cumplan** con la asistencia, quedarán en condición de **libres.**

Durante la exposición de los trabajos grupales, los docentes formularán preguntas a cada uno de los integrantes del grupo para completar su **calificación individual** en el trabajo.

# Parte I

# INTRODUCCIÓN A LAS BASES DE DATOS

Barberis, G. y Bombelli, E.

# Introducción a las bases de datos

Un *sistema de información* es el conjunto de recursos que permiten recoger, gestionar, controlar y difundir la información de un sistema u organización.

Un sistema de información está formado por los siguientes componentes:

- La base de datos.
- El sistema de gestión de bases de datos (SGBD).
- Los programas de aplicación.
- Los dispositivos físicos (ordenadores, medios de almacenamiento, etc.).
- El personal que utiliza y desarrolla el sistema.

La base de datos es un componente fundamental de un sistema de información. Las etapas para realizar una aplicación de bases de datos son las siguientes:

1. Planificación del proyecto.
2. Definición del sistema.
3. Recolección y análisis de los requisitos.
4. Diseño de la base de datos.
5. Selección del SGBD.
6. Diseño de la aplicación.
7. Prototipado.
8. Implementación.
9. Conversión y carga de datos.
10. Prueba.
11. Mantenimiento.

Estas etapas no son estrictamente secuenciales. De hecho hay que repetir algunas de las mismas varias veces, haciendo lo que se conoce como *ciclos de realimentación*. Por ejemplo, los problemas que se encuentran en la etapa de diseño de la base de datos, pueden requerir una recolección de requisitos adicional y su posterior análisis.

A continuación, se muestran las tareas más importantes que se realizan en cada etapa.

***1. Planificación del proyecto***

Esta etapa conlleva la planificación de cómo se pueden llevar a cabo las etapas de la manera más eficiente. Hay tres componentes principales: el trabajo que se ha de realizar, los recursos para llevarlo a cabo y el dinero para pagar por todo ello. Como apoyo a esta etapa, se necesitará un *modelo de datos* del sistema u organización que se quiera o necesite representar, en donde se muestren las entidades principales y sus relaciones, y en donde se identifiquen las principales áreas funcionales. Normalmente, este modelo de datos se representa mediante un

diagrama entidad-relación. En este modelo se tiene que mostrar también qué datos comparten las distintas áreas funcionales del sistema u organización.

La planificación de la base de datos también incluye el desarrollo de estándares que especifiquen cómo realizar la recolección de datos, cómo especificar su formato, qué documentación será necesaria y cómo se va a llevar a cabo el diseño y la implementación. El desarrollo y el mantenimiento de los estándares puede llevar bastante tiempo, pero si están bien diseñados, son una base para el personal informático en formación y para medir la calidad, además, garantizan que el trabajo se ajuste a patrones, independientemente de las habilidades y experiencia del diseñador. Por ejemplo, se pueden establecer reglas sobre cómo dar nombres a los datos, lo que evitará redundancias e inconsistencias. Se deben documentar todos los aspectos legales sobre los datos y los establecidos por el sistema u organización como, por ejemplo, qué datos deben tratarse de modo confidencial.

***2. Definición del sistema***

En esta etapa **se especifica el ámbito y los límites de la aplicación de bases de datos**, así como con qué otros sistemas interactúan. También hay que determinar quienes serán los usuarios y las áreas de aplicación.

***3. Recolección y análisis de los requisitos***

En esta etapa se recogen y analizan los requerimientos de los usuarios y de las áreas de aplicación. Esta información se puede recoger de varias formas:

- Entrevistando al personal de la organización, concretamente, a aquellos que son considerados expertos en las áreas de interés.
- Observando el funcionamiento de la organización o sistema.
- Examinando documentos, sobre todo aquellos que se utilizan para recoger o visualizar información.
- Utilizando cuestionarios para recoger información de grandes grupos de usuarios.
- Utilizando la experiencia adquirida en el diseño de sistemas similares.

La información recogida debe incluir las principales áreas de aplicación y los grupos de usuarios, la documentación utilizada o generada por estas áreas de aplicación o grupos de usuarios.

Esta etapa tiene como resultado un conjunto de documentos con las especificaciones de requisitos de los usuarios, en donde se describen las operaciones que se realizan en la organización desde distintos puntos de vista.

La información recogida se debe estructurar utilizando *técnicas de especificación de requisitos*, como por ejemplo técnicas de análisis y diseño estructurado y diagramas de flujo de datos.

***4. Diseño de la base de datos***

Esta etapa consta de tres fases: diseño conceptual, diseño lógico y diseño físico de la base de datos. La primera fase consiste en la producción de un esquema conceptual, que es independiente de todas las consideraciones físicas. Este modelo se refina después en un esquema lógico eliminando las construcciones que no se pueden representar en el modelo de base de datos escogido (relacional, orientado a objetos, etc.). En la tercera fase, el esquema lógico se traduce en un esquema físico para el SGBD escogido. La fase de diseño físico

considera las estructuras de almacenamiento y los métodos de acceso necesarios para proporcionar una llegada eficiente a la base de datos en memoria secundaria.

Los objetivos del diseño de la base de datos son:

- ξ Representar los datos que requieren las principales áreas de aplicación y representar las relaciones entre dichos datos.
- ξ Especificar un esquema que alcance las prestaciones requeridas para el sistema.

Hay varias estrategias a seguir para realizar el diseño: de abajo hacia arriba, de arriba hacia abajo, de adentro hacia afuera y la estrategia mixta. La estrategia *de abajo hacia arriba* parte de todos los atributos y los va agrupando en entidades y relaciones. Es apropiada cuando la base de datos es simple, con pocos atributos. La estrategia *de arriba hacia abajo* es más apropiada cuando se trata de bases de datos complejas. Se comienza con un esquema con entidades de alto nivel, que se van refinando para obtener entidades de bajo nivel, atributos y relaciones. La estrategia *de adentro hacia afuera* es similar a la estrategia de abajo hacia arriba, pero difiere en que se parte de los conceptos principales y se va extendiendo el esquema para considerar también otros conceptos, asociados con los que se han identificado en primer lugar. La estrategia *mixta* utiliza ambas estrategias, de abajo hacia arriba y de arriba hacia abajo, con un esquema del tipo *divide y vencerás*. Se obtiene un esquema inicial de alto nivel, se divide en partes, y de cada parte se obtiene un sub-esquema. Estos sub-esquemas se integran después para obtener el modelo final.

***5. Selección del SGBD***

Si no se dispone de un SGBD, o el que hay se encuentra obsoleto, se debe escoger un SGBD que sea adecuado para el sistema de información. Esta elección se debe hacer en cualquier momento, pero siempre antes del diseño lógico.

***6. Diseño de la aplicación***

En esta etapa se diseñan los programas de aplicación que usarán y procesarán la base de datos. Esta etapa y el diseño de la base de datos, son paralelas. En la mayor parte de los casos no se puede finalizar el diseño de las aplicaciones hasta que se ha terminado con el diseño de la base de datos. Por otro lado, la base de datos existe para dar soporte a las aplicaciones, por lo que habrá una realimentación desde el diseño de las aplicaciones al diseño de la base de datos.

***7. Prototipado***

Esta etapa, que es opcional, es para construir prototipos de la aplicación que permitan a los diseñadores y a los usuarios probar el sistema. Un prototipo es un modelo de trabajo de las aplicaciones del sistema.

***8. Implementación***

En esta etapa se crean las interfaces, se implementan los menús, los formularios para la introducción de datos y los informes de visualización de datos. Para ello, el SGBD puede disponer de lenguajes de cuarta generación que permiten el desarrollo rápido de aplicaciones mediante lenguajes de consultas no procedurales, generadores de informes, generadores de formularios, generadores de gráficos y generadores de aplicaciones.

***9. Conversión y carga de datos***

Esta etapa es necesaria cuando se está reemplazando un sistema antiguo por uno nuevo.

***10. Prueba***

Para llevar a cabo esta etapa, se debe haber decidido cuál es el SGBD que se va a utilizar, ya que el esquema físico se adapta a él. Entre el diseño físico y el diseño lógico hay una realimentación, ya que algunas de las decisiones que se tomen durante el diseño físico para mejorar las prestaciones, pueden afectar a la estructura del esquema lógico.

En general, el propósito del diseño físico es describir cómo se va a implementar físicamente el esquema lógico obtenido en la fase anterior. Concretamente, en el modelo relacional, esto consiste en:

ξ Obtener un conjunto de relaciones (tablas) y las restricciones que se deben cumplir sobre ellas.

ξ Determinar las estructuras de almacenamiento y los métodos de acceso que se van a utilizar para conseguir prestaciones óptimas.

### Diseño de interfaces de usuario

Antes de implementar los formularios y los informes, hay que diseñar su aspecto. Es conveniente tener en cuenta las siguientes recomendaciones:

ξ Utilizar títulos que sean significativos, que identifiquen sin ambigüedad el propósito del informe o formulario.

ξ Dar instrucciones breves y fáciles de comprender.

ξ Agrupar y secuenciar los campos de forma lógica.

ξ Hacer que el aspecto del informe o formulario sea atractivo a la vista.

ξ Utilizar nombres familiares para etiquetar los campos.

ξ Utilizar terminología y abreviaturas consistentes.

ξ Hacer un uso razonable y consistente de los colores.

ξ Dejar un espacio visible para los datos de entrada y delimitarlos.

ξ Permitir un uso sencillo y adecuado del cursor.

ξ Permitir la corrección carácter a carácter y de campos completos.

ξ Dar mensajes de error para los valores *ilegales*.

ξ Marcar los campos que sean opcionales.

ξ Dar mensajes a nivel de campo para explicar su significado.

ξ Dar una señal que indique cuándo el informe o formulario está completo.

# Diseño conceptual de bases de datos

## Modelos de datos

En el diseño de bases de datos se usan primero los modelos conceptuales para lograr una descripción de alto nivel de la realidad, y luego se transforma el esquema conceptual en un esquema lógico. El motivo de realizar estas dos etapas es la dificultad de abstraer la estructura de una base de datos que presente cierta complejidad. Un *esquema* es un conjunto de representaciones lingüísticas o gráficas que describen la estructura de los datos de interés.

Los modelos conceptuales deben ser buenas herramientas para representar la realidad, por lo que deben poseer las siguientes cualidades:

- *Expresividad*: Deben tener suficientes conceptos para expresar perfectamente la realidad.
- *Simplicidad*: Deben ser simples para que los esquemas sean fáciles de entender.
- *Minimalidad*: Cada concepto debe tener un significado distinto.
- *Formalidad*: Todos los conceptos deben tener una interpretación única, precisa y bien definida.

## El modelo entidad-relación

El modelo entidad-relación es el modelo conceptual más utilizado para el diseño conceptual de bases de datos. Fue introducido por Peter Chen en 1976. El modelo entidad-relación está formado por un conjunto de conceptos que permiten describir la realidad mediante un conjunto de representaciones gráficas y lingüísticas.

### *Entidad*

Es cualquier tipo de objeto o concepto sobre el que se recoge información: cosa, persona, concepto abstracto o suceso. Las entidades se representan gráficamente mediante rectángulos y su nombre aparece en el interior. Un nombre de entidad sólo puede aparecer una vez en el esquema conceptual.

Hay dos tipos de entidades: fuertes y débiles. Una *entidad débil* es una entidad cuya existencia depende de la existencia de otra entidad. Una *entidad fuerte* es una entidad que no es débil.

### *Relación (interrelación)*

Es una correspondencia o asociación entre dos o más entidades. Cada relación tiene un nombre que describe su función. Las relaciones se representan gráficamente mediante rombos y su nombre aparece en el interior.

Las entidades que están involucradas en una determinada relación se denominan *entidades participantes*. El número de participantes en una relación es lo que se denomina *grado* de la relación. Por lo tanto, una relación en la que participan dos entidades es una relación *binaria*; si son tres las entidades participantes, la relación es *ternaria*; etc.

Una *relación recursiva* es una relación donde la misma entidad participa más de una vez en la relación con distintos papeles. El nombre de estos papeles es importante para determinar la función de cada participación.

La *cardinalidad* con la que una entidad participa en una relación especifica el número mínimo y el número máximo de correspondencias en las que puede tomar parte cada ocurrencia de dicha entidad. La participación de una entidad en una relación es *obligatoria (total)* si la existencia de cada una de sus ocurrencias requiere la existencia de, al menos, una ocurrencia de la otra entidad participante. Si no, la participación es *opcional (parcial)*. Las reglas que definen la cardinalidad de las relaciones son las *reglas de una organización o sistema*.

A veces, surgen problemas cuando se está diseñado un esquema conceptual. Estos problemas, denominados *trampas*, suelen producirse a causa de una mala interpretación en el significado de alguna relación, por lo que es importante comprobar que el esquema conceptual carece de dichas trampas. En general, para encontrar las trampas, hay que asegurarse de que se entiende completamente el significado de cada relación. Si no se entienden las relaciones, se puede crear un esquema que no represente fielmente la realidad.

Una de las trampas que pueden encontrarse ocurre cuando el esquema representa una relación entre entidades, pero el camino entre algunas de sus ocurrencias es ambiguo. El modo de resolverla es reestructurando el esquema para representar la asociación entre las entidades correctamente.

Otra de las trampas sucede cuando un esquema sugiere la existencia de una relación entre entidades, pero el camino entre una y otra no existe para algunas de sus ocurrencias. En este caso, se produce una pérdida de información que se puede subsanar introduciendo la relación que sugería el esquema y que no estaba representada.

***Atributo***

Es una característica de interés o un hecho sobre una entidad o sobre una relación. Los atributos representan las propiedades básicas de las entidades y de las relaciones. Toda la información extensiva es portada por los atributos. Gráficamente, se representan mediante elipses que cuelgan de las entidades o relaciones a las que pertenecen.

Cada atributo tiene un conjunto de valores asociados denominado *dominio*. El dominio define todos los valores posibles que puede tomar un atributo. Puede haber varios atributos definidos sobre un mismo dominio.

Los atributos pueden ser simples o compuestos. Un *atributo simple* es un atributo que tiene un solo componente, que no se puede dividir en partes más pequeñas que tengan un significado propio. Un *atributo compuesto* es un atributo con varios componentes, cada uno con un significado por sí mismo.

Los atributos también pueden clasificarse en monovalentes o polivalentes. Un *atributo monovalente* es aquel que tiene un solo valor para cada ocurrencia de la entidad o relación a la que pertenece. Un *atributo polivalente* es aquel que tiene varios valores para cada ocurrencia de la entidad o relación a la que pertenece. A estos atributos también se les denomina *multivaluados*, y pueden tener un número máximo y un número mínimo de valores. La *cardinalidad* de un atributo indica el número mínimo y el número máximo de valores que puede tomar para cada ocurrencia de la entidad o relación a la que pertenece. El valor por omisión es uno a uno (1,1).

Por último, los atributos pueden ser derivados. Un *atributo derivado* es aquel que representa un valor que se puede obtener a partir del valor de uno o varios atributos, que no necesariamente deben pertenecer a la misma entidad o relación.

***Identificador***

Un identificador de una entidad es un atributo o conjunto de atributos que determina de modo único cada ocurrencia de esa entidad. Un identificador de una entidad debe cumplir dos condiciones:

1. No pueden existir dos ocurrencias de la entidad con el mismo valor del identificador.
2. Si se omite cualquier atributo del identificador, la condición anterior deja de cumplirse.

Toda entidad tiene al menos un identificador y puede tener varios identificadores alternativos. Las relaciones no tienen identificadores.

# Metodología de diseño conceptual

El primer paso en el diseño de una base de datos es la producción del esquema conceptual. El esquema conceptual también tendrá una documentación, que se irá produciendo durante su desarrollo. Las tareas a realizar en el diseño conceptual son las siguientes:

1. Identificar las entidades.
2. Identificar las relaciones.
3. Identificar los atributos y asociarlos a entidades y relaciones.
4. Determinar los dominios de los atributos.
5. Determinar los identificadores.
6. Determinar las jerarquías de generalización (si las hay).
7. Dibujar el diagrama entidad-relación.
8. Revisar el esquema conceptual local con el usuario.

***1. Identificar las entidades***

En primer lugar hay que definir los principales objetos que interesan al usuario. Estos objetos serán las entidades. Una forma de identificar las entidades es examinar las especificaciones de requisitos de usuario. En estas especificaciones se buscan los nombres que se mencionan (por ejemplo: número de empleado, nombre de empleado, número de inmueble, dirección del inmueble, alquiler, número de habitaciones). También se buscan objetos importantes como personas, lugares o conceptos de interés, excluyendo aquellos nombres que sólo son propiedades de otros objetos. Por ejemplo, se pueden agrupar el número de empleado y el nombre de empleado en una entidad denominada *empleado*, y agrupar número de inmueble, dirección del inmueble, alquiler y número de habitaciones en otra entidad denominada *inmueble*.

No siempre es obvio saber si un objeto es una entidad, una relación o un atributo. Por ejemplo ¿cómo se podría clasificar *matrimonio*? Pues de cualquiera de las tres formas. El análisis es subjetivo, por lo que distintos diseñadores pueden hacer distintas interpretaciones, aunque todas igualmente válidas. Todo depende de la opinión y la experiencia de cada uno. Los diseñadores de bases de datos deben tener una visión selectiva y clasificar las cosas que observan dentro del contexto de la empresa, organización o sistema. A partir de unas especificaciones de usuario es posible que no se pueda deducir un conjunto único de entidades, pero después de varias iteraciones del proceso de análisis, se llegará a obtener un conjunto de entidades que sean adecuadas para el sistema que se ha de construir.

Conforme se van identificando las entidades, se les dan nombres que tengan un significado y que sean obvias para el usuario. Los nombres de las entidades y sus descripciones se anotan en el diccionario de datos.

***2. Identificar las relaciones***

Una vez definidas las entidades, se deben definir las relaciones existentes entre ellas. Del mismo modo que para identificar las entidades se buscaban nombres en las especificaciones de requisitos, para identificar las relaciones se suelen buscar las expresiones verbales (por ejemplo: oficina tiene empleados, empleado gestiona inmueble, cliente visita inmueble). Si las

especificaciones de requisitos reflejan estas relaciones es porque son importantes para la empresa y, por lo tanto, se deben reflejar en el esquema conceptual.

Pero sólo interesan las relaciones que son necesarias. En el ejemplo anterior, se han identificado las relaciones *empleado gestiona inmueble* y *cliente visita inmueble*. Se podría pensar en incluir una relación entre empleado y cliente: *empleado atiende a cliente*, pero observando las especificaciones de requisitos no parece que haya interés en modelar tal relación.

La mayoría de las relaciones son binarias (entre dos entidades), pero no hay que olvidar que también puede haber relaciones en las que participen más de dos entidades, así como relaciones recursivas.

Es muy importante repasar las especificaciones para comprobar que todas las relaciones, explícitas o implícitas, se han encontrado. Si se tienen pocas entidades, se puede comprobar por parejas si hay alguna relación entre ellas.

Una vez identificadas todas las relaciones, hay que determinar la cardinalidad mínima y máxima con la que participa cada entidad en cada una de ellas. De este modo, el esquema representa de un modo más explícito la semántica de las relaciones. La cardinalidad es un tipo de restricción que se utiliza para comprobar y mantener la calidad de los datos. Estas restricciones son aseveraciones sobre las entidades que se pueden aplicar cuando se actualiza la base de datos para determinar si las actualizaciones violan o no las reglas establecidas sobre la semántica de los datos.

Conforme se van identificando las relaciones, se les van asignando nombres que tengan significado para el usuario. En el diccionario de datos se anotan los nombres de las relaciones, su descripción y las cardinalidades con las que participan las entidades en ellas.

***3. Identificar los atributos y asociarlos a entidades y relaciones***

Al igual que con las entidades, se buscan nombres en las especificaciones de requisitos. Son atributos los nombres que identifican propiedades, cualidades, identificadores o características de entidades o relaciones.

Lo más sencillo es preguntarse, para cada entidad y cada relación, ¿qué información se quiere saber de ...? La respuesta a esta pregunta se debe encontrar en las especificaciones de requisitos. Pero, en ocasiones, será necesario preguntar a los usuarios para que aclaren los requisitos. Desgraciadamente, los usuarios pueden dar respuestas a esta pregunta que también contengan otros conceptos, por lo que hay que considerar sus respuestas con mucho cuidado.

Al identificar los atributos, hay que tener en cuenta si son simples o compuestos. Por ejemplo, el atributo *dirección* puede ser simple, teniendo la dirección completa como un solo valor: Rivadavia 4500, Bs. As.; o puede ser un atributo compuesto, formado por la *calle* (Rivadavia), el *número* (4500) y la *localidad* (Bs. As.). El escoger entre atributo simple o compuesto depende de los requisitos del usuario. Si el usuario no necesita acceder a cada uno de los componentes de la dirección por separado, se puede representar como un atributo simple. Pero si el usuario quiere acceder a los componentes de forma individual, entonces se debe representar como un atributo compuesto.

También se deben identificar los atributos derivados o calculados, que son aquellos cuyo valor se puede calcular a partir de los valores de otros atributos. Por ejemplo, el número de empleados de cada oficina, la edad de los empleados o el número de inmuebles que gestiona cada empleado. Algunos diseñadores no representan los atributos derivados en los esquemas

conceptuales. Si se hace, se debe indicar claramente que el atributo es derivado y a partir de qué atributos se obtiene su valor. Donde hay que considerar los atributos derivados es en el diseño físico.

Cuando se están identificando los atributos, se puede descubrir alguna entidad que no se ha identificado previamente, por lo que hay que volver al principio introduciendo esta entidad y viendo si se relaciona con otras entidades.

Es muy útil elaborar una lista de atributos e ir eliminándolos de la lista conforme se vayan asociando a una entidad o relación. De este modo, uno se puede asegurar de que cada atributo se asocia a una sola entidad o relación, y que cuando la lista se ha acabado, se han asociado todos los atributos.

Hay que tener mucho cuidado cuando parece que un mismo atributo se debe asociar a varias entidades. Esto puede ser por una de las siguientes causas:

- Se han identificado varias entidades, como *director*, *supervisor* y *administrativo*, cuando, de hecho, pueden representarse como una sola entidad denominada *empleado*. En este caso, se puede escoger entre introducir una jerarquía de generalización, o dejar las entidades que representan cada uno de los puestos.
- Se ha identificado una relación entre entidades. En este caso, se debe asociar el atributo a una sola de las entidades y hay que asegurarse de que la relación ya se había identificado previamente. Si no es así, se debe actualizar la documentación para recoger la nueva relación.

Conforme se van identificando los atributos, se les asignan nombres que tengan significado para el usuario. De cada atributo se debe anotar la siguiente información:

- Nombre y descripción del atributo.
- Alias o sinónimos por los que se conoce al atributo.
- Tipo de dato y longitud.
- Valores por defecto del atributo (si se especifican).
- Si el atributo siempre va a tener un valor (si admite o no nulos).
- Si el atributo es compuesto y, en su caso, qué atributos simples lo forman.
- Si el atributo es derivado y, en su caso, cómo se calcula su valor.
- Si el atributo es multi-valuado.

#### *4. Determinar los dominios de los atributos*

El dominio de un atributo es el conjunto de valores que puede tomar el atributo. Un esquema conceptual está completo si incluye los dominios de cada atributo; los valores permitidos para cada atributo, su tamaño y su formato. También se puede incluir información adicional sobre los dominios como, por ejemplo, las operaciones que se pueden realizar sobre cada atributo, qué atributos pueden compararse entre sí o qué atributos pueden combinarse con otros. Aunque sería muy interesante que el sistema final respetara todas estas indicaciones sobre los dominios, esto es todavía una línea abierta de investigación. Toda la información sobre los dominios se debe anotar también en el diccionario de datos.

### *5. Determinar los identificadores*

Cada entidad tiene al menos un identificador. En este paso, se trata de encontrar todos los identificadores de cada una de las entidades. Los identificadores pueden ser simples o compuestos. De cada entidad se escogerá uno de los identificadores como clave primaria en la fase del diseño lógico.

Cuando se determinan los identificadores es fácil darse cuenta si una entidad es fuerte o débil. Si una entidad tiene al menos un identificador, es *fuerte* (otras denominaciones son *padre*, *propietaria* o *dominante*). Si una entidad no tiene atributos que le sirvan de identificador, es *débil* (otras denominaciones son *hijo*, *dependiente* o *subordinada*).

Todos los identificadores de las entidades se deben anotar en el diccionario de datos.

### *6. Determinar las jerarquías de generalización*

En este paso hay que observar las entidades que se han identificado hasta el momento. Hay que ver si es necesario reflejar las diferencias entre distintas ocurrencias de una entidad, con lo que surgirán nuevas sub-entidades de esta entidad genérica; o bien, si hay entidades que tienen características en común y que realmente son sub-entidades de una nueva entidad genérica.

### *7. Dibujar el diagrama entidad-relación*

Una vez identificados todos los conceptos, se puede dibujar el diagrama entidad-relación correspondiente a una de las vistas de los usuarios. Se obtiene así un esquema conceptual local.

# Diseño lógico de bases de datos

El objetivo del diseño lógico es convertir los esquemas conceptuales locales en un esquema lógico global que se ajuste al modelo de SGBD sobre el que se vaya a implementar el sistema. Mientras que el objetivo fundamental del diseño conceptual es la complexión y expresividad de los esquemas conceptuales locales, el objetivo del diseño lógico es obtener una representación que use, del modo más eficiente posible, los recursos que el modelo de SGBD posee para estructurar los datos y para modelar las restricciones.

Los modelos de bases de datos más extendidos son el modelo relacional, el modelo de red y el modelo jerárquico. El modelo orientado a objetos es también muy popular, pero no existe un modelo estándar orientado a objetos.

El modelo relacional (y los modelos previos) carecen de ciertos rasgos de abstracción que se usan en los modelos conceptuales. Por lo tanto, un primer paso en la fase del diseño lógico consistirá en la conversión de esos mecanismos de representación de alto nivel en términos de las estructuras de bajo nivel disponibles en el modelo relacional.

## Metodología de diseño lógico en el modelo relacional

La metodología que se va a seguir para el diseño lógico en el modelo relacional consta de dos fases, cada una de ellas compuesta por varios pasos que se detallan a continuación.

- Construir y validar los esquemas lógicos locales para cada vista de usuario.
    1. Convertir los esquemas conceptuales locales en esquemas lógicos locales.
    2. Derivar un conjunto de relaciones (tablas) para cada esquema lógico local.
    3. Validar cada esquema mediante la normalización.
    4. Validar cada esquema frente a las transacciones del usuario.
    5. Dibujar el diagrama entidad-relación.
    6. Definir las restricciones de integridad.
    7. Revisar cada esquema lógico local con el usuario correspondiente.
- Construir y validar el esquema lógico global.
    1. Mezclar los esquemas lógicos locales en un esquema lógico global.
    2. Validar el esquema lógico global.
    3. Estudiar el crecimiento futuro.
    4. Dibujar el diagrama entidad-relación final.
    5. Revisar el esquema lógico global con los usuarios.

En la primera fase, se construyen los esquemas lógicos locales para cada vista de usuario y se validan. En esta fase se refinan los esquemas conceptuales creados durante el diseño conceptual, eliminando las estructuras de datos que no se pueden implementar de manera

directa sobre el modelo que soporta el SGBD, en el caso que nos ocupa, el modelo relacional. Una vez hecho esto, se obtiene un primer esquema lógico que se valida mediante la normalización y frente a las transacciones que el sistema debe llevar a cabo, tal y como se refleja en las especificaciones de requisitos de usuario. El esquema lógico ya validado se puede utilizar como base para el desarrollo de prototipos. Una vez finalizada esta fase, se dispone de un esquema lógico para cada vista de usuario que es correcto, comprensible y sin ambigüedad.

***1. Convertir los esquemas conceptuales locales en esquemas lógicos locales***

En este paso, se eliminan de cada esquema conceptual las estructuras de datos que los sistemas relacionales no modelan directamente:

a) *Eliminar las relaciones de muchos a muchos,* sustituyendo cada una de ellas por una nueva entidad intermedia y dos relaciones de uno a muchos de esta nueva entidad con las entidades originales. La nueva entidad será débil, ya que sus ocurrencias dependen de la existencia de ocurrencias en las entidades originales.

b) *Eliminar las relaciones entre tres o más entidades,* sustituyendo cada una de ellas por una nueva entidad (débil) intermedia que se relaciona con cada una de las entidades originales. La cardinalidad de estas nuevas relaciones binarias dependerá de su significado.

c) *Eliminar las relaciones recursivas,* sustituyendo cada una de ellas por una nueva entidad (débil) y dos relaciones binarias de esta nueva entidad con la entidad original. La cardinalidad de estas relaciones dependerá de su significado.

d) *Eliminar las relaciones con atributos,* sustituyendo cada una de ellas por una nueva entidad (débil) y las relaciones binarias correspondientes de esta nueva entidad con las entidades originales. La cardinalidad de estas relaciones dependerá del tipo de la relación original y de su significado.

e) *Eliminar los atributos multi-valuados,* sustituyendo cada uno de ellos por una nueva entidad (débil) y una relación binaria de uno a muchos con la entidad original.

f) *Revisar las relaciones de uno a uno,* ya que es posible que se hayan identificado dos entidades que representen el mismo objeto (sinónimos). Si así fuera, ambas entidades deben integrarse en una sola.

g) *Eliminar las relaciones redundantes.* Una relación es redundante cuando se puede obtener la misma información que ella aporta mediante otras relaciones. El hecho de que haya dos caminos diferentes entre dos entidades no implica que uno de los caminos corresponda a una relación redundante, eso dependerá del significado de cada relación.

Una vez finalizado este paso, es más correcto referirse a los esquemas conceptuales locales refinados como esquemas lógicos locales, ya que se adaptan al modelo de base de datos que soporta el SGBD escogido.

***2. Derivar un conjunto de relaciones (tablas) para cada esquema lógico local***

En este paso, se obtiene un conjunto de relaciones (tablas) para cada uno de los esquemas lógicos locales en donde se representen las entidades y relaciones entre entidades, que se describen en cada una de las vistas que los usuarios tienen del sistema. Cada relación de la base de datos tendrá un nombre, y el nombre de sus atributos aparecerá, a continuación, entre paréntesis. El atributo o atributos que forman la clave primaria se subrayan. Las claves ajenas,

mecanismo que se utiliza para representar las relaciones entre entidades en el modelo relacional, se especifican aparte indicando la relación (tabla) a la que hacen referencia.

A continuación, se describe cómo las relaciones (tablas) del modelo relacional representan las entidades y relaciones que pueden aparecer en los esquemas lógicos.

a) *Entidades fuertes.* Crear una relación para cada entidad fuerte que incluya todos sus atributos simples. De los atributos compuestos incluir sólo sus componentes.

Cada uno de los identificadores de la entidad será una clave candidata. De entre las claves candidatas hay que escoger la clave primaria; el resto serán claves alternativas. Para escoger la clave primaria entre las claves candidatas se pueden seguir estas indicaciones:

- Escoger la clave candidata que tenga menos atributos.
- Escoger la clave candidata cuyos valores no tengan probabilidad de cambiar en el futuro.
- Escoger la clave candidata cuyos valores no tengan probabilidad de perder la unicidad en el futuro.
- Escoger la clave candidata con el mínimo número de caracteres (si es de tipo texto).
- Escoger la clave candidata más fácil de utilizar desde el punto de vista de los usuarios.

b) *Entidades débiles.* Crear una relación para cada entidad débil incluyendo todos sus atributos simples. De los atributos compuestos incluir sólo sus componentes. Añadir una clave ajena a la entidad de la que depende. Para ello, se incluye la clave primaria de la relación que representa a la entidad padre en la nueva relación creada para la entidad débil. A continuación, determinar la clave primaria de la nueva relación.

c) *Relaciones binarias de uno a uno.* Para cada relación binaria se incluyen los atributos de la clave primaria de la entidad padre en la relación (tabla) que representa a la entidad hijo, para actuar como una clave ajena. La entidad hijo es la que participa de forma total (obligatoria) en la relación, mientras que la entidad padre es la que participa de forma parcial (opcional). Si las dos entidades participan de forma total o parcial en la relación, la elección de padre e hijo es arbitraria. Además, en caso de que ambas entidades participen de forma total en la relación, se tiene la opción de integrar las dos entidades en una sola relación (tabla). Esto se suele hacer si una de las entidades no participa en ninguna otra relación.

d) *Relaciones binarias de uno a muchos.* Como en las relaciones de uno a uno, se incluyen los atributos de la clave primaria de la entidad padre en la relación (tabla) que representa a la entidad hijo, para actuar como una clave ajena. Pero ahora, la entidad padre es la de ``la parte del muchos" (cada padre tiene muchos hijos), mientras que la entidad hijo es la de ``la parte del uno" (cada hijo tiene un solo padre).

e) *Jerarquías de generalización.* En las jerarquías, se denomina entidad padre a la entidad genérica y entidades hijo a las sub-entidades. Hay tres opciones

distintas para representar las jerarquías. La elección de la más adecuada se hará en función de su tipo (total/parcial, exclusiva/superpuesta).

1. Crear una relación por cada entidad. Las relaciones de las entidades hijo heredan como clave primaria la de la entidad padre. Por lo tanto, la clave primaria de las entidades hijo es también una clave ajena al padre. Esta opción sirve para cualquier tipo de jerarquía, total o parcial y exclusiva o superpuesta.

2. Crear una relación por cada entidad hijo, heredando los atributos de la entidad padre. Esta opción sólo sirve para jerarquías totales y exclusivas.

3. Integrar todas las entidades en una relación, incluyendo en ella los atributos de la entidad padre, los atributos de todos los hijos y un atributo discriminativo para indicar el caso al cual pertenece la entidad en consideración. Esta opción sirve para cualquier tipo de jerarquía. Si la jerarquía es superpuesta, el atributo discriminativo será multi-valuado.

Una vez obtenidas las relaciones con sus atributos, claves primarias y claves ajenas, sólo queda actualizar el diccionario de datos con los nuevos atributos que se hayan identificado en este paso.

***3. Validar cada esquema mediante la normalización***

La normalización se utiliza para mejorar el esquema lógico, de modo que satisfaga ciertas restricciones que eviten la duplicidad de datos. La normalización garantiza que el esquema resultante se encuentra más próximo al modelo de la empresa, sistema u organización, que es consistente y que tiene la mínima redundancia y la máxima estabilidad.

La normalización es un proceso que permite decidir a qué entidad pertenece cada atributo. Uno de los conceptos básicos del modelo relacional es que los atributos se agrupan en relaciones (tablas) porque están relacionados a nivel lógico. En la mayoría de las ocasiones, una base de datos normalizada no proporciona la máxima eficiencia, sin embargo, el objetivo ahora es conseguir una base de datos normalizada por las siguientes razones:

- Un esquema normalizado organiza los datos de acuerdo a sus dependencias funcionales, es decir, de acuerdo a sus relaciones lógicas.

- El esquema lógico no tiene porqué ser el esquema final. Debe representar lo que el diseñador entiende sobre la naturaleza y el significado de los datos de lo que se quiera modelar. Si se establecen unos objetivos en cuanto a prestaciones, el diseño físico cambiará el esquema lógico de modo adecuado. Una posibilidad es que algunas relaciones normalizadas se desnormalicen. Pero la desnormalización no implica que se haya malgastado tiempo normalizando, ya que mediante este proceso el diseñador aprende más sobre el significado de los datos. De hecho, la normalización obliga a entender completamente cada uno de los atributos que se han de representar en la base de datos.

- Un esquema normalizado es robusto y carece de redundancias, por lo que está libre de ciertas anomalías que éstas pueden provocar cuando se actualiza la base de datos.

- Los equipos informáticos de hoy en día son mucho más potentes, por lo que en ocasiones es más razonable implementar bases de datos fáciles de manejar (las normalizadas), a costa de un tiempo adicional de proceso.

- La normalización produce bases de datos con esquemas flexibles que pueden extenderse con facilidad.

El objetivo de este paso es obtener un conjunto de relaciones que se encuentren en la forma normal de Boyce-Codd. Para ello, hay que pasar por la primera, segunda y tercera formas normales.

### *4. Validar cada esquema frente a las transacciones del usuario*

El objetivo de este paso es validar cada esquema lógico local para garantizar que puede soportar las transacciones requeridas por los correspondientes usuarios. Estas transacciones se encontrarán en las especificaciones de requisitos de usuario. Lo que se debe hacer es tratar de realizar las transacciones de forma manual utilizando el diagrama entidad-relación, el diccionario de datos y las conexiones que establecen las claves ajenas de las relaciones (tablas). Si todas las transacciones se pueden realizar, el esquema queda validado. Pero si alguna transacción no se puede realizar, seguramente será porque alguna entidad, relación o atributo no se ha incluido en el esquema.

### *5. Dibujar el diagrama entidad-relación*

En este momento, se puede dibujar el diagrama entidad-relación final para cada vista de usuario que recoja la representación lógica de los datos desde su punto de vista. Este diagrama habrá sido validado mediante la normalización y frente a las transacciones de los usuarios.

### *6. Definir las restricciones de integridad*

Las restricciones de integridad son reglas que se quieren imponer para proteger la base de datos, de modo que no pueda llegar a un estado inconsistente. Hay cinco tipos de restricciones de integridad.

a) *Datos requeridos.* Algunos atributos deben contener valores en todo momento, es decir, no admiten nulos.

b) *Restricciones de dominios.* Todos los atributos tienen un dominio asociado, que es el conjunto de los valores que cada atributo puede tomar.

c) *Integridad de entidades.* El identificador de una entidad no puede ser nulo, por lo tanto, las claves primarias de las relaciones (tablas) no admiten nulos.

d) *Integridad referencial.* Una clave ajena enlaza cada tupla de la relación hijo con la tupla de la relación padre que tiene el mismo valor en su clave primaria. La integridad referencial dice que si una clave ajena tiene un valor (si es no nula), ese valor debe ser uno de los valores de la clave primaria a la que referencia. Hay varios aspectos a tener en cuenta sobre las claves ajenas para lograr que se cumpla la integridad referencial.

   1. ¿Admite nulos la clave ajena? Cada clave ajena expresa una relación. Si la participación de la entidad hijo en la relación es total, entonces la clave ajena no admite nulos; si es parcial, la clave ajena debe aceptar nulos.
   2. ¿Qué hacer cuando se quiere borrar una ocurrencia de la entidad padre que tiene algún hijo? O lo que es lo mismo, ¿qué hacer cuando se quiere borrar una tupla que está siendo referenciada por otra tupla a través de una clave ajena? Hay varias respuestas posibles:
      - *Restringir:* No se pueden borrar tuplas que están siendo referenciadas por otras tuplas.
      - *Propagar:* Se borra la tupla deseada y se propaga el borrado a todas las tuplas que le hacen referencia.

- *Anular:* Se borra la tupla deseada y todas las referencias que tenía se ponen, automáticamente, a nulo (esta respuesta sólo es válida si la clave ajena acepta nulos).
- *Valor por defecto:* Se borra la tupla deseada y todas las referencias toman, automáticamente, el valor por defecto (esta respuesta sólo es válida si se ha especificado un valor por defecto para la clave ajena).
- *No comprobar:* Se borra la tupla deseada y no se hace nada para garantizar que se siga cumpliendo la integridad referencial.

3. ¿Qué hacer cuando se quiere modificar la clave primaria de una tupla que está siendo referenciada por otra tupla a través de una clave ajena? Las respuestas posibles son las mismas que en el caso anterior. Cuando se escoge propagar, se actualiza la clave primaria en la tupla deseada y se propaga el cambio a los valores de clave ajena que le hacían referencia.

e) *Reglas de la organización.* Cualquier operación que se realice sobre los datos debe cumplir las restricciones que impone el funcionamiento de lo que se esta modelando.

Todas las restricciones de integridad establecidas en este paso se deben reflejar en el diccionario de datos para que puedan ser tenidas en cuenta durante la fase del diseño físico.

### *7. Revisar cada esquema lógico local con el usuario correspondiente*

Para garantizar que cada esquema lógico local es una fiel representación de la vista del usuario, lo que se debe hacer es comprobar con él es que lo reflejado en el esquema y en la documentación es correcto y está completo.

*Relación entre el esquema lógico y los diagramas de flujo de datos*

El esquema lógico refleja la estructura de los datos a almacenar que maneja la empresa, sistema u organización. Un diagrama de flujo de datos muestra cómo se mueven los datos en la empresa, sistema u organización y los sitios de almacenamiento en donde se guardan. Si se han utilizado diagramas de flujo de datos para modelar las especificaciones de requisitos de usuario, se pueden utilizar para comprobar la consistencia y completitud del esquema lógico desarrollado. Para ello:

- Cada almacén de datos debe corresponder con una o varias entidades completas.
- Los atributos en los flujos de datos deben corresponder a alguna entidad.

Los esquemas lógicos locales obtenidos hasta este momento se integrarán en un solo esquema lógico global en la siguiente fase para modelar los datos de todo el sistema.

### *8. Mezclar los esquemas lógicos locales en un esquema lógico global*

En este paso, se deben integrar todos los esquemas locales en un solo esquema global. En un sistema pequeño, con dos o tres vistas de usuario y unas pocas entidades y relaciones, es relativamente sencillo comparar los esquemas locales, mezclarlos y resolver cualquier tipo de diferencia que pueda existir. Pero en los sistemas grandes, se debe seguir un proceso más sistemático para llevar a cabo este paso con éxito:

1. Revisar los nombres de las entidades y sus claves primarias.

2. Revisar los nombres de las relaciones.
3. Mezclar las entidades de las vistas locales.
4. Incluir (sin mezclar) las entidades que pertenecen a una sola vista de usuario.
5. Mezclar las relaciones de las vistas locales.
6. Incluir (sin mezclar) las relaciones que pertenecen a una sola vista de usuario.
7. Comprobar que no se ha omitido ninguna entidad ni relación.
8. Comprobar las claves ajenas.
9. Comprobar las restricciones de integridad.
10. Dibujar el esquema lógico global.
11. Actualizar la documentación.

***9. Validar el esquema lógico global***

Este proceso de validación se realiza, de nuevo, mediante la normalización y mediante la prueba frente a las transacciones de los usuarios. Pero ahora sólo hay que normalizar las relaciones que hayan cambiado al mezclar los esquemas lógicos locales y sólo hay que probar las transacciones que requieran acceso a áreas que hayan sufrido algún cambio.

***10. Estudiar el crecimiento futuro***

En este paso, se trata de comprobar que el esquema obtenido puede acomodar los futuros cambios en los requisitos con un impacto mínimo. Si el esquema lógico se puede extender fácilmente, cualquiera de los cambios previstos se podrá incorporar al mismo con un efecto mínimo sobre los usuarios existentes.

***11. Dibujar el diagrama entidad-relación final***

Una vez validado el esquema lógico global, ya se puede dibujar el diagrama entidad-relación que representa el modelo de los datos de la empresa que son de interés. La documentación que describe este modelo (incluyendo el esquema relacional y el diccionario de datos) se debe actualizar y completar.

***12. Revisar el esquema lógico global con los usuarios***

Una vez más, se debe revisar con los usuarios el esquema global y la documentación obtenida para asegurarse de que son una fiel representación de la empresa, sistema u organización.

# Normalización

La normalización es una técnica para diseñar la estructura lógica de los datos de un sistema de información en el modelo relacional, desarrollada por E. F. Codd en 1972. Es una estrategia de diseño de abajo hacia arriba, se parte de los atributos y éstos se van agrupando en relaciones (tablas) según su afinidad. **Aquí no se utilizará la normalización como una técnica de diseño de bases de datos, sino como una etapa posterior a la correspondencia entre el esquema conceptual y el esquema lógico**, que elimine las dependencias entre atributos no deseadas. Las ventajas de la normalización son las siguientes:

- Evita anomalías en inserciones, modificaciones y borrados.
- Mejora la independencia de datos.
- No establece restricciones artificiales en la estructura de los datos.

Uno de los conceptos fundamentales en la normalización es el de *dependencia funcional*. Una dependencia funcional es una relación entre atributos de una misma relación (tabla). La dependencia funcional es una noción semántica. Si hay o no dependencias funcionales entre atributos no lo determina una serie abstracta de reglas, sino, más bien, los modelos mentales del usuario y las reglas de funcionamiento de la organización para la que se desarrolla el sistema de información. Cada dependencia funcional es una clase especial de regla de integridad y representa una relación de uno a muchos.

En el proceso de normalización se debe ir comprobando que cada relación (tabla) cumple una serie de reglas que se basan en la clave primaria y las dependencias funcionales. Cada regla que se cumple aumenta el grado de normalización. Si una regla no se cumple, la relación se debe descomponer en varias relaciones que sí la cumplan.

La normalización se lleva a cabo en una serie pasos. Cada paso corresponde a una forma normal que tiene unas propiedades. Conforme se va avanzando en la normalización, las relaciones tienen un formato más estricto (más fuerte) y, por lo tanto, son menos vulnerables a las anomalías de actualización. El modelo relacional sólo requiere un conjunto de relaciones en primera forma normal. Las restantes formas normales son opcionales. Sin embargo, para evitar las anomalías de actualización, es recomendable llegar al menos a la tercera forma normal.

### *Primera forma normal (1FN)*

Una relación está en primera forma normal si, y sólo si, todos los dominios de la misma contienen valores atómicos, es decir, no hay grupos repetitivos. Si se ve la relación gráficamente como una tabla, estará en 1FN si tiene un solo valor en la intersección de cada fila con cada columna.

Si una relación no está en 1FN, hay que eliminar de ella los grupos repetitivos. Un grupo repetitivo será el atributo o grupo de atributos que tiene múltiples valores para cada tupla de la relación. Hay dos formas de eliminar los grupos repetitivos. En la primera, se repiten los atributos con un solo valor para cada valor del grupo repetitivo. De este modo, se introducen redundancias ya que se duplican valores, pero estas redundancias se eliminarán después mediante las restantes formas normales. La segunda forma de eliminar los grupos repetitivos consiste en poner cada uno de ellos en una relación aparte, heredando la clave primaria de la relación en la que se encontraban.

Un conjunto de relaciones se encuentra en 1FN si ninguna de ellas tiene grupos repetitivos.

*Segunda forma normal (2FN)*

Una relación está en segunda forma normal si, y sólo si, está en 1FN y, además, cada atributo que no está en la clave primaria es completamente dependiente de la clave primaria.

La 2FN se aplica a las relaciones que tienen claves primarias compuestas por dos o más atributos. Si una relación está en 1FN y su clave primaria es simple (tiene un solo atributo), entonces también está en 2FN. Las relaciones que no están en 2FN pueden sufrir anomalías cuando se realizan actualizaciones.

Para pasar una relación de 1FN a 2FN hay que eliminar las dependencias parciales de la clave primaria. Para ello, se eliminan los atributos que son funcionalmente dependientes y se ponen en una nueva relación con una copia de su determinante (los atributos de la clave primaria de los que dependen).

*Tercera forma normal (3FN)*

Una relación está en tercera forma normal si, y sólo si, está en 2FN y, además, cada atributo que no está en la clave primaria no depende transitivamente de la clave primaria.

Aunque las relaciones en 2FN tienen menos redundancias que las relaciones en 1FN, todavía pueden sufrir anomalías frente a las actualizaciones. Para pasar una relación de 2FN a 3FN hay que eliminar las dependencias transitivas. Para ello, se eliminan los atributos que dependen transitivamente y se ponen en una nueva relación con una copia de su determinante (el atributo o atributos no clave de los que dependen).

***Forma normal de Boyce-Codd (BCFN o 4FN)***

Una relación está en la forma normal de Boyce-Codd si, y sólo si, todo determinante es una clave candidata.

La 2FN y la 3FN eliminan las dependencias parciales y las dependencias transitivas de la clave primaria. Pero este tipo de dependencias todavía pueden existir sobre otras claves candidatas, si éstas existen. La BCFN es más fuerte que la 3FN, por lo tanto, toda relación en BCFN está en 3FN.

La violación de la BCFN es poco frecuente ya que se da bajo ciertas condiciones que raramente se presentan. Se debe comprobar si una relación viola la BCFN si tiene dos o más claves candidatas compuestas que tienen al menos un atributo en común.

# Diseño físico de bases de datos

Mientras que en el diseño lógico se especifica qué se guarda, en el diseño físico se especifica cómo se guarda. Para ello, el diseñador debe conocer muy bien toda la funcionalidad del SGBD concreto que se vaya a utilizar y también el sistema informático sobre el que éste va a trabajar. El diseño físico no es una etapa aislada, ya que algunas decisiones que se tomen durante su desarrollo, por ejemplo para mejorar las prestaciones, pueden provocar una reestructuración del esquema lógico.

## Metodología de diseño físico para bases de datos relacionales

El objetivo de esta etapa es producir una descripción de la implementación de la base de datos en memoria secundaria. Esta descripción incluye las estructuras de almacenamiento y los métodos de acceso a datos que se utilizarán para lograr eficiencia en los mismos.

El diseño físico se divide de cuatro fases, cada una de ellas compuesta por una serie de pasos:

- Traducir el esquema lógico global para el SGBD específico
    1. Diseñar las relaciones base para el SGBD específico.
    2. Diseñar las reglas de sistema para el SGBD específico.
- Diseñar la representación física.
    1. Analizar las transacciones.
    2. Escoger las organizaciones de archivos.
    3. Escoger los índices secundarios.
    4. Considerar la introducción de redundancias controladas.
    5. Estimar la necesidad de espacio en disco.
- Diseñar los mecanismos de seguridad.
    1. Diseñar las vistas de los usuarios.
    2. Diseñar las reglas de acceso.
- Monitorear y refinar el sistema.

## Traducir el esquema lógico global

La primera fase del diseño lógico consiste en traducir el esquema lógico global en un esquema que se pueda implementar en el SGBD escogido. Para ello, es necesario conocer toda la funcionalidad que éste ofrece. Por ejemplo, el diseñador deberá saber:

- Si el sistema soporta la definición de claves primarias, claves ajenas y claves alternativas.
- Si el sistema soporta la definición de datos requeridos (es decir, si se pueden definir atributos como no nulos).

- *Red.* La red se convierte en un cuello de botella cuando tiene mucho tráfico y cuando hay muchas colisiones.

Cada uno de estos recursos afecta a los demás, de modo que una mejora en alguno de ellos puede provocar mejoras en otros.

***1. Analizar las transacciones***

Para realizar un buen diseño físico es necesario conocer las consultas y las transacciones que se van a ejecutar sobre la base de datos. Esto incluye tanto información cualitativa, como cuantitativa. Para cada transacción, hay que especificar:

- La frecuencia con que se va a ejecutar.
- Las relaciones y los atributos a los que accede la transacción, y el tipo de acceso: Consulta, inserción, modificación o eliminación. Los atributos que se modifican no son buenos candidatos para construir estructuras de acceso.
- Los atributos que se utilizan en los predicados del WHERE de las sentencias SQL. Estos atributos pueden ser candidatos para construir estructuras de acceso dependiendo del tipo de predicado que se utilice.
- Si es una consulta, los atributos involucrados en el *join* de dos o más relaciones. Estos atributos pueden ser candidatos para construir estructuras de acceso.
- Las restricciones temporales impuestas sobre la transacción. Los atributos utilizados en los predicados de la transacción pueden ser candidatos para construir estructuras de acceso.

***2. Escoger las organizaciones de ficheros***

El objetivo de este paso es escoger la organización de archivos óptima para cada relación. Por ejemplo, un archivo desordenado es una buena estructura cuando se va a cargar gran cantidad de datos en una relación al inicializarla, cuando la relación tiene pocas tuplas, también cuando en cada acceso se deben obtener todas las tuplas de la relación, o cuando la relación tiene una estructura de acceso adicional, como puede ser un índice. Por otra parte, los archivos dispersos (*hashing*) son apropiados cuando se accede a las tuplas a través de los valores exactos de alguno de sus campos (condición de igualdad en el WHERE). Si la condición de búsqueda es distinta de la igualdad (búsqueda por rango, por patrón, etc.), la dispersión no es una buena opción. Hay otras organizaciones, como la ISAM o los árboles B+.

Las organizaciones de archivos elegidas deben documentarse, justificando en cada caso la opción escogida.

***3. Escoger los índices secundarios***

Los índices secundarios permiten especificar caminos de acceso adicionales para las relaciones base. Pero hay que tener en cuenta que estos índices conllevan un costo de mantenimiento que hay que sopesar frente a la ganancia en prestaciones. A la hora de seleccionar los índices, se pueden seguir las siguientes indicaciones:

- Construir un índice sobre la clave primaria de cada relación base.
- No crear índices sobre relaciones pequeñas.
- Añadir un índice sobre los atributos que se utilizan para acceder con mucha frecuencia.

- Añadir un índice sobre las claves ajenas que se utilicen con frecuencia para hacer *joins*.
- Evitar los índices sobre atributos que se modifican a menudo.
- Evitar los índices sobre atributos poco selectivos (aquellos en los que la consulta selecciona una porción significativa de la relación).
- Evitar los índices sobre atributos formados por tiras de caracteres largas.

Los índices creados se deben documentar, explicando las razones de su elección.

***4. Considerar la introducción de redundancias controladas***

En ocasiones puede ser conveniente relajar las reglas de normalización introduciendo redundancias de forma controlada, con objeto de mejorar las prestaciones del sistema. En la etapa del diseño lógico se recomienda llegar, al menos, hasta la tercera forma normal para obtener un esquema con una estructura consistente y sin redundancias. Pero, a menudo, sucede que las bases de datos así normalizadas no proporcionan la máxima eficiencia, con lo que es necesario volver atrás y desnormalizar algunas relaciones, sacrificando los beneficios de la normalización para mejorar las prestaciones. Es importante hacer notar que la desnormalización sólo debe realizarse cuando se estime que el sistema no puede alcanzar las prestaciones deseadas. Y, desde luego, la necesidad de desnormalizar en ocasiones no implica eliminar la normalización del diseño lógico: la normalización obliga al diseñador a entender completamente cada uno de los atributos que se han de representar en la base de datos. Por lo tanto, hay que tener en cuenta los siguientes factores:

- La desnormalización hace que la implementación sea más compleja.
- La desnormalización hace que se sacrifique la flexibilidad.
- La desnormalización puede hacer que los accesos a datos sean más rápidos, pero demora las actualizaciones.

Por regla general, la desnormalización de una relación puede ser una opción viable cuando las prestaciones que se obtienen no son las deseadas y la relación se actualiza con poca frecuencia, pero se consulta muy a menudo. Las redundancias que se pueden incluir al desnormalizar son de varios tipos: se pueden introducir datos derivados (calculados a partir de otros datos), se pueden duplicar atributos o se pueden hacer *joins* de relaciones.

El incluir un atributo derivado dependerá del costo adicional de almacenarlo y mantenerlo consistente con los datos de los que se deriva, frente al costo de calcularlo cada vez que se necesita.

No se pueden establecer una serie de reglas que determinen cuándo desnormalizar relaciones, pero hay algunas situaciones muy comunes en donde puede considerarse esta posibilidad:

- *Combinar relaciones uno a uno.* Cuando hay relaciones (tablas) involucradas en relaciones uno a uno, se accede a ellas de manera conjunta con frecuencia y prácticamente no se accede de manera separada; se pueden combinar en una sola relación (tabla).
- *Duplicar atributos no clave en relaciones uno a muchos para reducir los joins.* Para evitar operaciones de *join*, se pueden incluir atributos de la relación (tabla) padre en la relación (tabla) hijo de las relaciones uno a muchos.

ξ *Tablas de referencia.* Las tablas de referencia (*lookup*) son listas de valores, cada uno de los cuales tiene un código. Este tipo de tablas son un caso de relación uno a muchos. De este modo, es muy fácil validar los datos, además de que se ahorra espacio escribiendo sólo el código y no la descripción para cada caso, además de ahorrar tiempo cuando se actualizan las descripciones. Si las tablas de referencia se utilizan a menudo en consultas críticas, se puede considerar la introducción de la descripción junto con el código en la relación (tabla) hijo, manteniendo la tabla de referencia para validación de datos.

ξ *Duplicar claves ajenas en relaciones uno a muchos para reducir los joins.* Para evitar operaciones de *join*, se pueden incluir claves ajenas de una relación (tabla) en otra relación (tabla) con la que se relaciona (habrá que tener en cuenta ciertas restricciones).

ξ *Duplicar atributos en relaciones muchos a muchos para reducir los joins.* Durante el diseño lógico se eliminan las relaciones muchos a muchos, introduciendo dos relaciones uno a muchos. Esto hace que aparezca una nueva relación (tabla) intermedia, de modo que si se quiere obtener la información de la relación muchos a muchos, se tiene que realizar el *join* de tres relaciones (tablas). Para evitar algunos de estos *joins* se pueden incluir algunos de los atributos de las relaciones (tablas) originales en la relación (tabla) intermedia.

ξ *Introducir grupos repetitivos.* Los grupos repetitivos se eliminan en el primer paso de la normalización para conseguir la primera forma normal. Estos grupos se eliminan introduciendo una nueva relación (tabla), generando una relación uno a muchos. A veces, puede ser conveniente reintroducir los grupos repetitivos para mejorar las prestaciones.

Todas las redundancias que se introduzcan en este paso se deben documentar y razonar. El esquema lógico se debe actualizar para reflejar los cambios introducidos.

***5. Estimar la necesidad de espacio en disco***

En caso de que se tenga que adquirir nuevo equipamiento informático, el diseñador debe estimar el espacio necesario en disco para la base de datos. Esta estimación depende del SGBD que se vaya a utilizar y del hardware. En general, se debe estimar el número de tuplas de cada relación y su tamaño. También se debe estimar el factor de crecimiento de cada relación.

## Diseñar los mecanismos de seguridad

Los datos constituyen un recurso esencial para cualquier organización, por lo tanto su seguridad es de vital importancia. Durante el diseño lógico se habrán especificado los requerimientos en cuanto a seguridad que en esta fase se deben implementar. Para llevar a cabo esta implementación, el diseñador debe conocer las posibilidades que ofrece el SGBD que se vaya a utilizar.

***1. Diseñar las vistas de los usuarios***

El objetivo de este paso es diseñar las vistas de los usuarios correspondientes a los esquemas lógicos locales. Las vistas, además de preservar la seguridad, mejoran la independencia de datos, reducen la complejidad y permiten que los usuarios vean los datos en el formato deseado.

***2. Diseñar las reglas de acceso***

El administrador de la base de datos asigna a cada usuario un identificador que tendrá una palabra secreta asociada por motivos de seguridad. Para cada usuario o grupo de usuarios se otorgarán permisos para realizar determinadas acciones sobre determinados objetos de la base de datos.

## Monitorear y refinar el sistema

Una vez implementado el esquema físico de la base de datos, se debe poner en marcha para observar sus prestaciones. Si éstas no son las deseadas, el esquema deberá cambiar para intentar satisfacerlas. Una vez refinado el esquema, no permanecerá estático, ya que tendrá que ir cambiando conforme lo requieran los nuevos requisitos de los usuarios. Los SGBD proporcionan herramientas para monitorear el sistema mientras está en funcionamiento.

# Ejercicio comentado

## Análisis de densidad poblacional

Se le ha contratado con el objeto de realizar un manejo ecológico de la recolección y eliminación de residuos del conurbano bonaerense. Es por ello que uno de los primeros pasos consiste en determinar cuál es la densidad poblacional, por lo que su análisis se basará en los municipios existentes, las viviendas y las personas dentro de los municipios.

Se parte del supuesto que cada persona sólo puede habitar en una vivienda y pertenecer a un municipio, pero puede ser propietaria de varias viviendas. Interesa saber, por otro lado, el destino de las viviendas, que podrá ser familiar o comercial. Por último, se indicarán los supuestos semánticos que se consideren oportunos para justificar las decisiones de diseño.

Es importante recalcar, que se debe conocer también cuál es la superficie de cada municipio, de cada vivienda, es decir superficie cubierta, así como la del terreno ocupado, ubicación de la vivienda, etc.

## Primer paso (diseño conceptual)

Si recordamos los pasos por los que debemos pasar, en lo que a metodología para el diseño de bases de datos se refiere, lo primero que debemos hacer es realizar una identificación de las entidades. Así realizamos un análisis de lo expresado en el enunciado y dividiendo el mismo en partes para simplificarlo, satisfaciendo las necesidades planteadas, podemos ver que hay tres candidatos a ser entidades que surgen del enunciado:

- ξ Municipio.
- ξ Vivienda.
- ξ Persona.

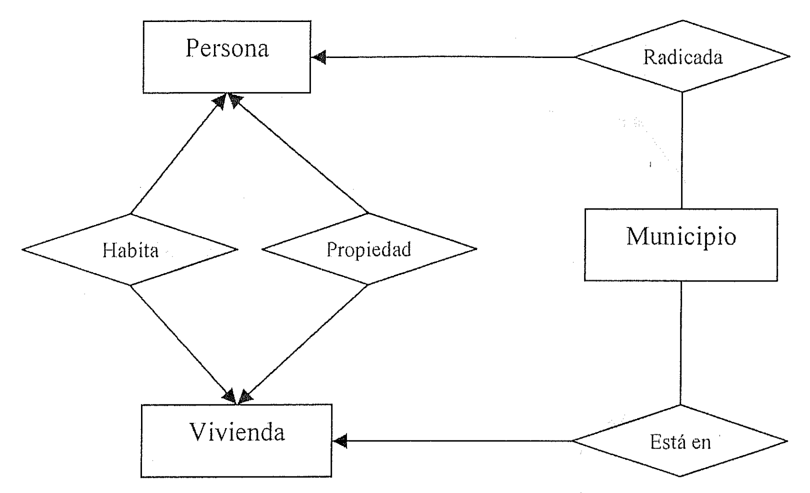
Otro aspecto es analizar las relaciones, teniendo en cuenta los supuestos mencionados en el enunciado inicial, encontrando que:

- ③ Cada persona solo puede habitar en una vivienda.
- ③ Cada persona puede ser propietaria de más de una vivienda.
- ③ Una persona puede estar radicada en un solo municipio.

Otros supuestos no enunciados, que surgen de la lógica son:

- ③ En una vivienda puede habitar más de una persona.
- ③ Más de una persona puede ser propietaria de una vivienda.
- ③ Un municipio puede tener varias viviendas.
- ③ Una vivienda solo puede estar en un municipio.

En base a esto desarrollamos un primer esquema:



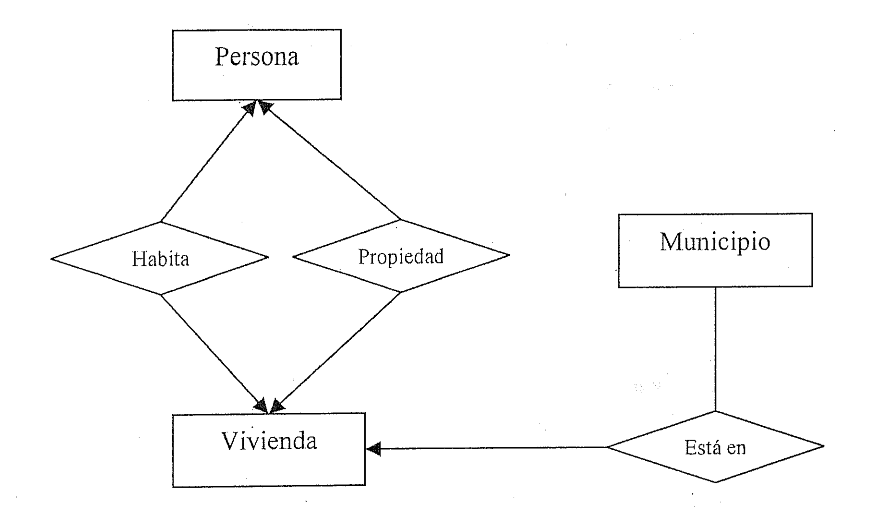

Analizaremos ahora, como son las interrelaciones:

③ Como dijimos, una persona habita como mínimo en una vivienda y en una vivienda puede que no habite ninguna persona o bien muchas.
③ Cada persona puede ser propietaria de más de una vivienda y una vivienda puede tener más de un propietario.
③ Una vivienda está en un solo municipio y un municipio puede tener muchas viviendas.
③ Algo similar ocurre con la interrelación que tenemos entre las personas, y su radicación en los distintos municipios.

Ahora analizaremos si existen redundancias. Esto lo haremos buscando si existe alguna interrelación que pueda obtenerse a partir de otras interrelaciones.

③ Si analizamos la interrelación **Propiedad**, la misma no puede ser explicada por "Está en" y "Radicada", ya que el hecho de que sea propietaria no quiere decir que la habite y que resida en un municipio.
③ Si intentáramos eliminar la interrelación **Habita**, la misma no puede ser explicada por "Está en" y "Radicada", ya que mediante "Radicada" sabemos las personas que están en el municipio y con "Está en" las viviendas dentro del municipio (con "Propiedad" dijimos que no necesariamente habita en la misma), pero no podemos saber que personas habitan en cada vivienda. Por ello "Habita" no se puede eliminar.
③ En cuanto a la interrelación **Está en**, tampoco puede ser explicada por las otras interrelaciones, ya que con "Habita" sabemos las personas que hay en una casa y con "Radicada" las personas que están en un municipio, pero puede suceder que tengamos casas en las que no tengamos ninguna persona. Por ello no se puede eliminar.
③ Por último, **Radicada;** en este caso con "Habita" sabemos las personas que están en una vivienda y con "Está en" sabemos en que municipio está esa vivienda, por lo que aún sacando la palabra "Radicada" sabremos las personas que tenemos en un municipio, por lo que esta interrelación si puede ser eliminada.

De esta forma nos queda el siguiente esquema:




Para terminar nuestro esquema Entidad/Relación, en el que ya tenemos determinadas las entidades y sus interrelaciones, nos falta colocar los atributos de cada una de las entidades.

Del enunciado podemos sacar algunos atributos, que pueden visualizarse en forma directa. También tendremos otros que saldrán de los objetivos planteados en dicha introducción como ser:

En **VIVIENDA**:

- ③ Destino.
- ③ Ubicación.
- ③ Superficie cubierta.
- ③ Superficie del terreno.
- ③ Identificación de la vivienda (número de partida).

En **MUNICIPIO**:

- ③ Superficie total (surge del supuesto lógico de la vida real, que no toda la superficie del municipio está ocupada con viviendas).
- ③ Superficie de espacios públicos (este dato junto con el de superficie total y la suma de superficies del terreno, nos permitirán determinar cuál es la posibilidad de expansión del mismo).
- ③ Denominación que lo identifique.
- ③ Ubicación geográfica.

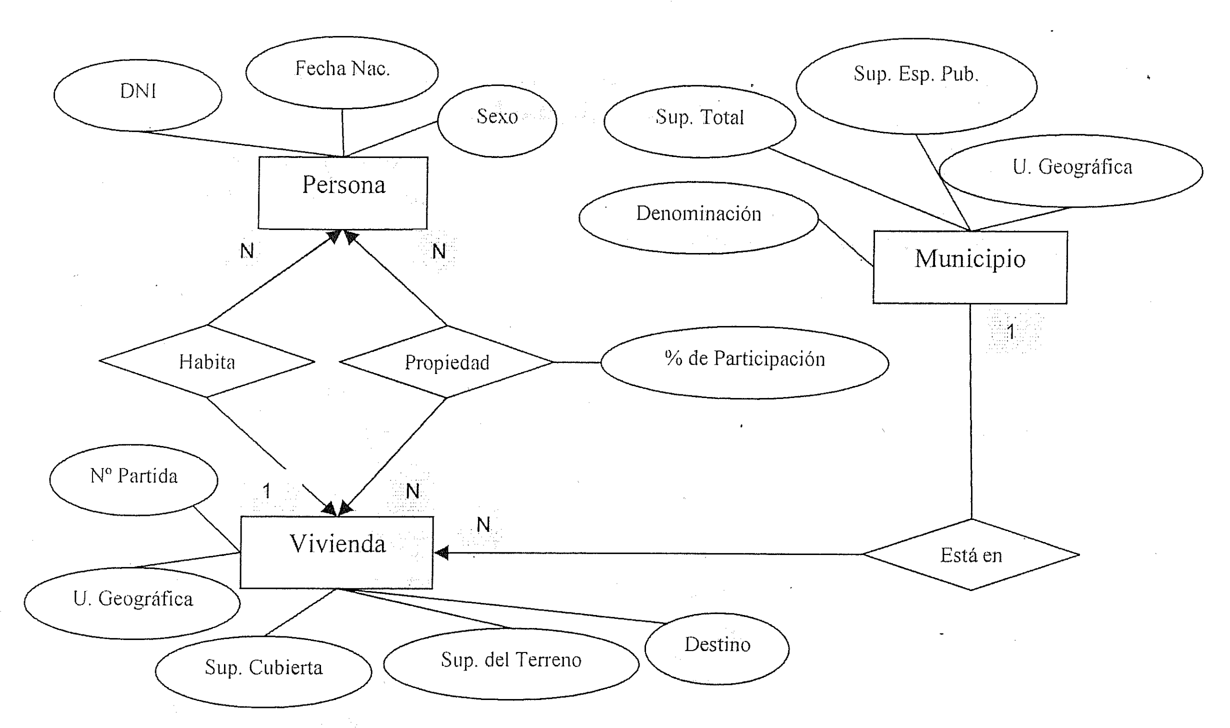
En **PERSONA**:

- ③ Fecha de Nacimiento (lo que nos dará la edad).
- ③ DNI que lo identifique.
- ③ Sexo.

En la relación **PROPIEDAD**:

- ③ % de Participación.

De esta manera hemos obtenido un esquema conceptual local que quedará como sigue:




## Segundo paso (diseño lógico)

Vamos ahora a convertir el esquema conceptual local en el correspondiente esquema lógico local.

En este paso, se eliminan del esquema conceptual, las estructuras de datos que los sistemas relacionales no modelan directamente. Para ello tendremos en cuenta los siguientes tópicos:

③ *Eliminar las relaciones muchos a muchos.*
③ *Eliminar las relaciones entre tres o más entidades.*
③ *Eliminar las relaciones recursivas.*
③ *Eliminar las relaciones con atributos.*
③ *Eliminar los atributos multi-valuados.*
③ *Revisar las relaciones uno a uno.*

Recordemos que para realizar el pasaje del esquema conceptual al esquema lógico debemos:

- ξ En primer lugar toda entidad se transforma en una tabla o relación.
- ξ Todo atributo se transforma en una columna (campo) dentro de una tabla.
- ξ El identificador único de la entidad se convierte en clave primaria.
- ξ Para las transformaciones de relaciones 1:N se utiliza la propagación de la clave, es decir propagamos el campo o atributo principal de la tabla que tiene cardinalidad máxima 1 a la que tiene cardinalidad máxima N. Desaparece de esta forma el nombre de la relación.

Analicemos las relaciones que tenemos:

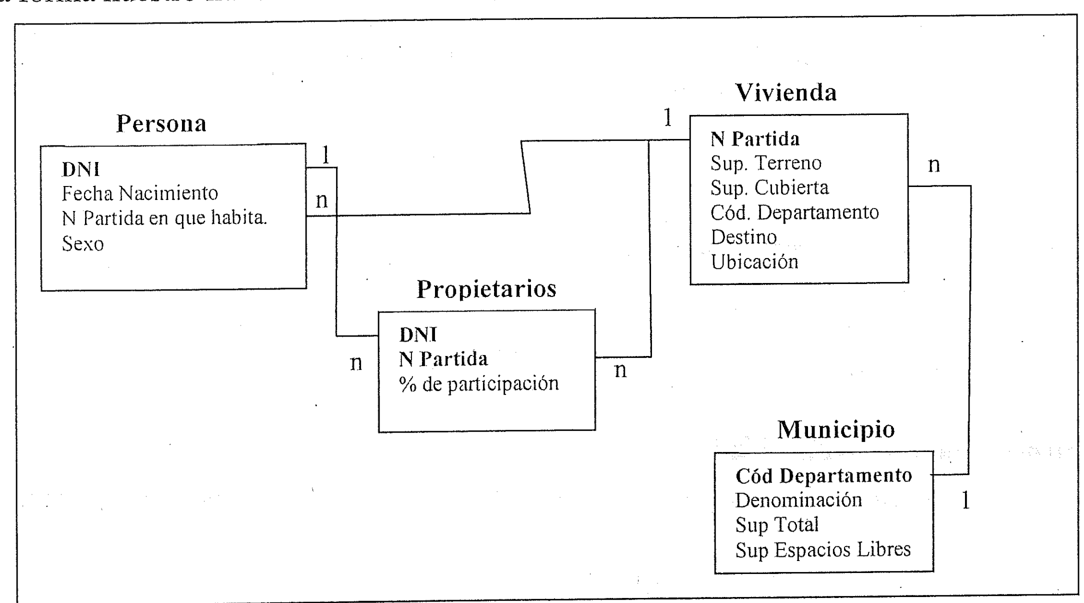
③ 1 Persona puede habitar en 1 Vivienda y 1 Vivienda puede ser habitada por muchas Personas, por ello la relación es 1 (vivienda) a muchos (persona). Aquí debemos entonces propagar la clave es decir en Nº Partida para saber donde habita esa persona.

③ 1 Persona puede ser Propietaria de muchas Viviendas, y una Vivienda puede tener como Propietarias muchas Personas. La relación es muchos a muchos. Es por ello que debemos crear una nueva entidad que estará conformada por ambas entidades y a la que llamaremos PROPIETARIOS, la que tendrá como atributos identificadores los de cada una de las

entidades madres, y como atributo propio, el porcentaje (%) de participación de cada persona propietaria (atributo que ya tenía la relación en el diseño físico).

③ Por último, una Vivienda puede estar en un solo Municipio y un Municipio puede contener muchas Viviendas, por lo que la relación es uno a muchos. Aquí también debemos propagar la clave, es decir Cód. Departamento pasa la tabla Vivienda.

De esta forma nuestro nuevo diseño será:




Queda así terminado nuestro esquema lógico, que en la notación propia del mismo sería:

Persona (**DNI**, N Partida en que habita, Sexo)
Propietarios (**DNI, N Partida,** % Participación)
Vivienda (**N Partida,** Sup Terreno, Sup Cubierta, Cód Departamento, Destino, Ubicación)
Municipio (**Cód Departamento,** Denominación, Sup Total, Sup Espacios Libres)

Para terminar, deberíamos someter dicho esquema lógico a las reglas de Normalización.

## Tercer paso

Un último paso a dar, para facilitar el proceso de construcción de nuestro esquema físico, es la creación de una planilla por entidad, en la que pondremos cada uno de los atributos y determinaremos que tipo de datos cargaremos (texto, numérico, fecha, etc) su tamaño, y una breve descripción del mismo.

**PERSONA**

| Nombre | Tipo | Tamaño | Descripción |
|---|---|---|---|
| DNI | Texto | 8 | El número sin puntos. |
| Numero de Partida en el que habita | Texto | 10 | Es el número de partida. |
| Fecha de Nacimiento | Fecha | | Con este campo podremos determinar la edad. |
| Sexo | Texto | 1 | Colocar un lista desplegable para que sea M o F. |

**PROPIEDADES**

| Nombre | Tipo | Tamaño | Descripción |
|---|---|---|---|
| DNI | Texto | 8 | El número sin puntos. |
| N Partida | Texto | 10 | Es el número de partida. |

| Porcentaje de participación | Número | % con 2 decimales | Podremos limitar este punto de manera que esté entre 0 y 100. |
|---|---|---|---|

VIVIENDA

| Nombre | Tipo | Tamaño | Descripción |
|---|---|---|---|
| N° Partida | Texto | 10 | Es el número de partida. |
| Sup. Terreno | Numérico | Decimal con 2 decimales | La superficie a consignar aquí es para el caso de edificios. |
| Sup. Cubierta | Numérico | Decimal con 2 decimales | |
| Cód. Dep./Partido | Texto | 5 | Se desprende del Cód. dado por el INDEC, donde los primeros dos dígitos son la Provincia y los tres siguientes son del Dep/Loc en si[1]. |
| Destino | Texto | 15 | Colocar un lista desplegable para que sea Familiar o Multifamiliar u Otros. |
| Ubicación | Texto | 12 | Seis dígitos para la Latitud y Seis para la Longitud. |

MUNICIPIO

| Nombre | Tipo | Tamaño | Descripción |
|---|---|---|---|
| Cód. Dep./Partido | Texto | 5 | Ídem a lo dicho en la tabla anterior. |
| Denominación | Texto | 20 | Es el Nombre del de Partido/ Departamento. |
| Superficie total | Numérico | Decimal con 2 decimales | |
| Superficie Esp. Libres | Numérico | Decimal con 2 decimales | |

# Bibliografía bases de datos


Batini C., Ceri S., Navathe S. B. (1994). *Diseño Conceptual de Bases de Datos. Un enfoque de entidades-interrelaciones* Addison-Wesley / Díaz de Santos.

Connolly T., Begg C., Strachan A. (1996). Database Systems. A Practical Approach to Design, Implementation and Management Addison-Wesley Segunda Edición en 1998.

Date C. J. (1993). Introducción a los Sistemas de Bases de Datos Volumen I, Quinta Edición Addison-Wesley Iberoamericana Sexta Edición en 1995 (en inglés, por Addison-Wesley).

Elmasri R., Navathe S. B. (1997). Sistemas de Bases de Datos. Conceptos fundamentales Segunda Edición Addison-Wesley Iberoamericana Tercera Edición en 1999 (en inglés, por Addison-Wesley).

Folk M. J., Zoellick B. (1992). File Structures Segunda Edición por Addison-Wesley.


[1] Según el INDEC cada localidad tiene un código que está compuesto por OCHO (8) dígitos, leído de izquierda a derecha. los DOS (2) primeros dígitos corresponden a división político-territorial (Provincias y Ciudad Autónoma de Buenos Aires), los TRES (3) dígitos subsiguientes corresponden a división político-administrativa (partidos, departamentos y distritos escolares para la Ciudad de Buenos Aires) y los TRES (3) últimos dígitos se refieren a la localidad propiamente dicha. Más información en http://www.indec.mecon.ar/principal.asp?id_tema=6205.

Hansen G. W., Hansen J. V. (1997). Diseño y Administración de Bases de Datos Segunda Edición, Prentice Hall.

Hernández M. J. (1997). Database Design for Mere Mortals Addison-Wesley Developers Press.

# Parte II

# MODELOS DE SIMULACIÓN

Bartoloni, N.; Cipriotti, P.A.; Garibaldi, L.; Grigera, G.;

Guerschman, J.P.; Perelman, S.B.; Sznaider, G.

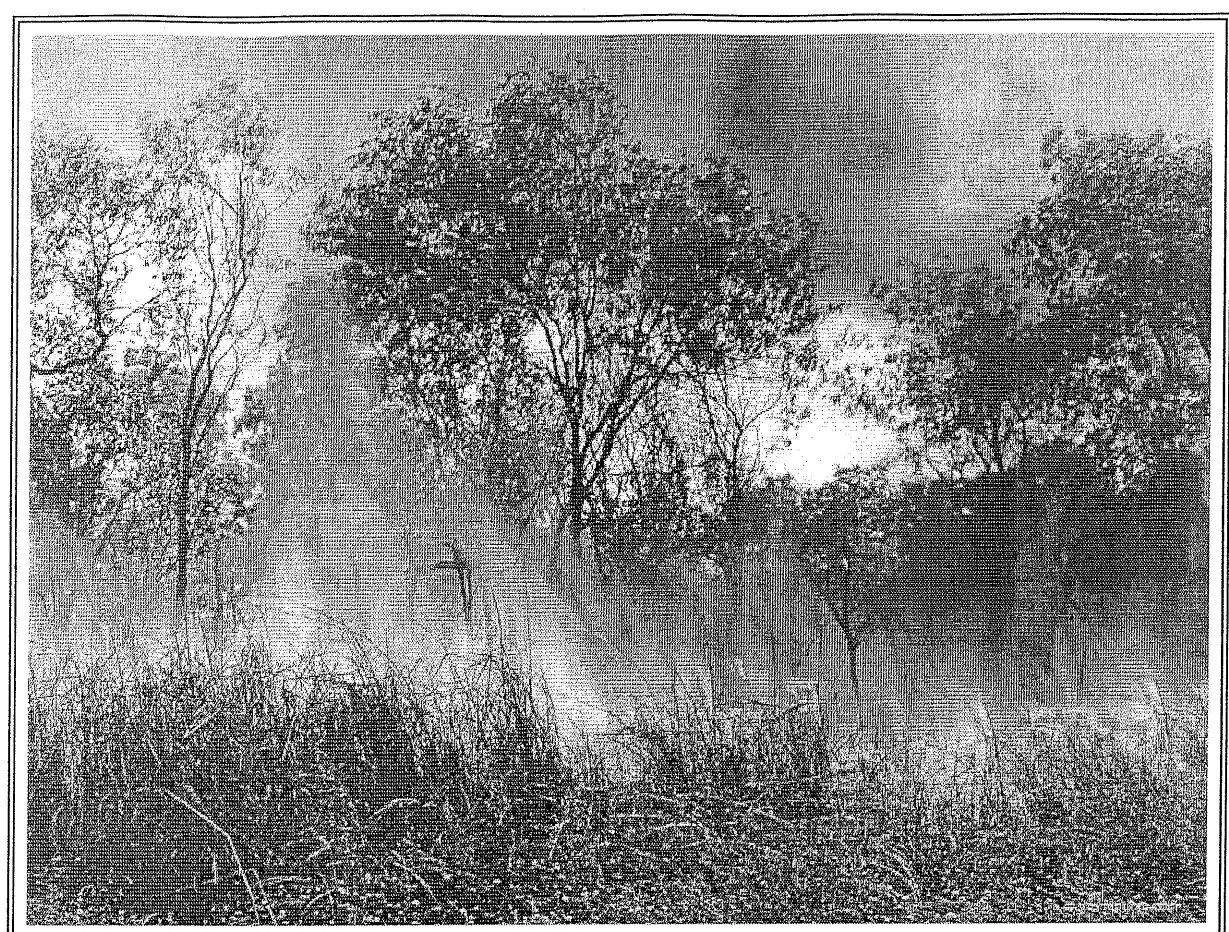
# Modelos de simulación en ciencias ambientales

## ¿Por qué modelos en ciencias ambientales?

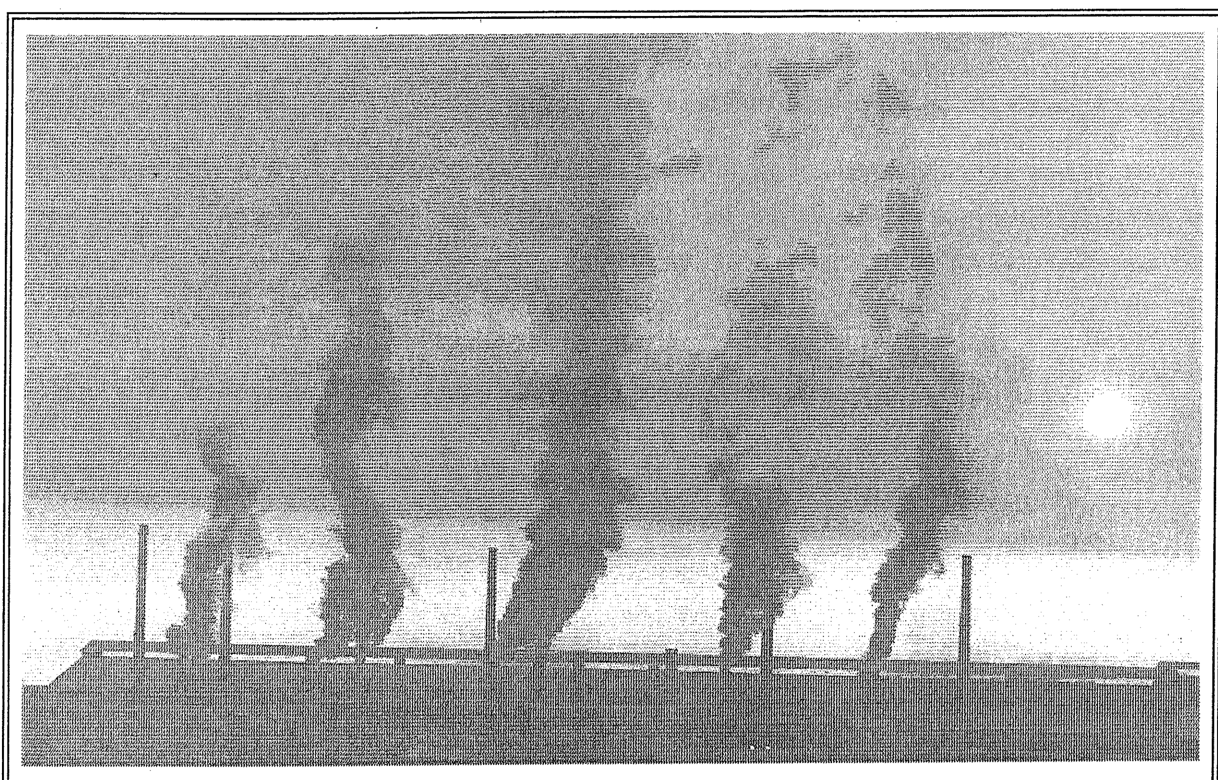
Los profesionales de las Ciencias Ambientales necesitan interpretar el funcionamiento de los sistemas ambientales y adquirir capacidad para explorar el impacto de diversas medidas sobre el estado futuro del sistema. Los sistemas ambientales están compuestos por múltiples partes interactuantes y generalmente presentan gran complejidad. En las fotos siguientes se ilustran una serie de situaciones ambientales de interés actual. Los modelos matemáticos han sido extensivamente usados para el análisis de sistemas de este tipo, pues permiten extraer de un sistema sus partes y relaciones teóricas básicas y explorar la evolución de ambos mediante ejercicios de simulación. Los resultados y conclusiones de estos ejercicios pueden ser extrapolados a los sistemas y permitir así la estimación de estados futuros y la toma de decisiones.



El Río de la Plata descarga en el océano Atlántico aguas turbias procedentes de los ríos Paraná y Uruguay. El Río de la Plata ha moldeado los sistemas naturales, económicos y sociales en el sur de Sudamérica. Los modelos ecológicos han sido usados para estudiar los sistemas naturales y para analizar los impactos de las actividades humanas en esta región.



Los incendios ocurren de manera natural en muchos ecosistemas de bosques y pastizales. ¿Deben ser suprimidos estos incendios cuando interfieren con los intereses del hombre? Los modelos basados en programas de computación han sido usados para brindar una respuesta a esta pregunta.



El uso de combustibles fósiles como el petróleo, gas o carbón ha producido un aumento de la concentración de CO2 en la atmósfera, lo que incrementa el llamado "efecto invernadero". Esto estaría provocando cambios en clima global, lo que tendrá consecuencias muy importantes sobre los ecosistemas y las actividades humanas. Los modelos de simulación son una herramienta muy valiosa para predecir los probables cambios en clima, como así también sus impactos sobre los ecosistemas.

# ¿Qué es modelar?

Cuando hablamos de modelar o modelado, hacemos referencia a la disciplina que se dedica a construir modelos. Un **modelo** es una abstracción o simplificación de un **sistema** a partir de la formalización del conocimiento (Figura 1). Un sistema es cualquier fenómeno que tiene al menos dos componentes diferenciables y que presentan algún grado de interacción entre sí. Esta definición es suficientemente amplia como para incluir muchos fenómenos de la vida cotidiana bajo el rótulo de sistema: desde un teléfono celular hasta el sistema eléctrico de un edificio moderno, un bosque o una represa hidroeléctrica completa. En la naturaleza encontramos sistemas con distinto grado de **complejidad**, y cada componente del sistema puede ser considerado un sistema en sí mismo. Por esto, cualquier sistema que deseamos estudiar forma parte de una **jerarquía** de otros sistemas y es en este punto donde el objetivo o problema particular a modelar establece los **límites** temporales, espaciales y conceptuales de un modelo.

Los modelos que actualmente conocemos pueden ser modelos físicos o digitales, por ejemplo un globo terráqueo es un modelo físico del planeta Tierra, del mismo modo que una maqueta es un modelo físico de un edificio. En cambio, existen modelos digitales que son representaciones virtuales de sistemas a través de lenguajes de programación o finalmente relaciones numéricas. Por ejemplo los juegos de guerra o grupos anti-terroristas en red actualmente muy difundidos, como el Counter Strike, son modelos digitales que representan una guerra real, del mismo modo que el conocido juego SimCity de los '80 simula el desarrollo de una ciudad. En este sentido, es importante destacar que así como poseemos los conocimientos y herramientas para simular una guerra o una ciudad, también podemos simular cualquier ecosistema real.

## Los lenguajes de programación

Los modelos arriba mencionados funcionan en computadoras al igual que tantos otros que veremos durante el curso. Para que funcionen necesitan que la computadora posea una especificación de las acciones a tomar (eventos al accionar teclas, controles, gráficos a mostrar, cálculos, etc.). La especificación se logra mediante lo que se denomina programa informático. Un programa es un listado de instrucciones, reglas o métodos que definen el comportamiento de la computadora y el modelo que se pretende representar. Los lenguajes de programación se pueden diferenciar, entre otras cosas, por su nivel de abstracción.

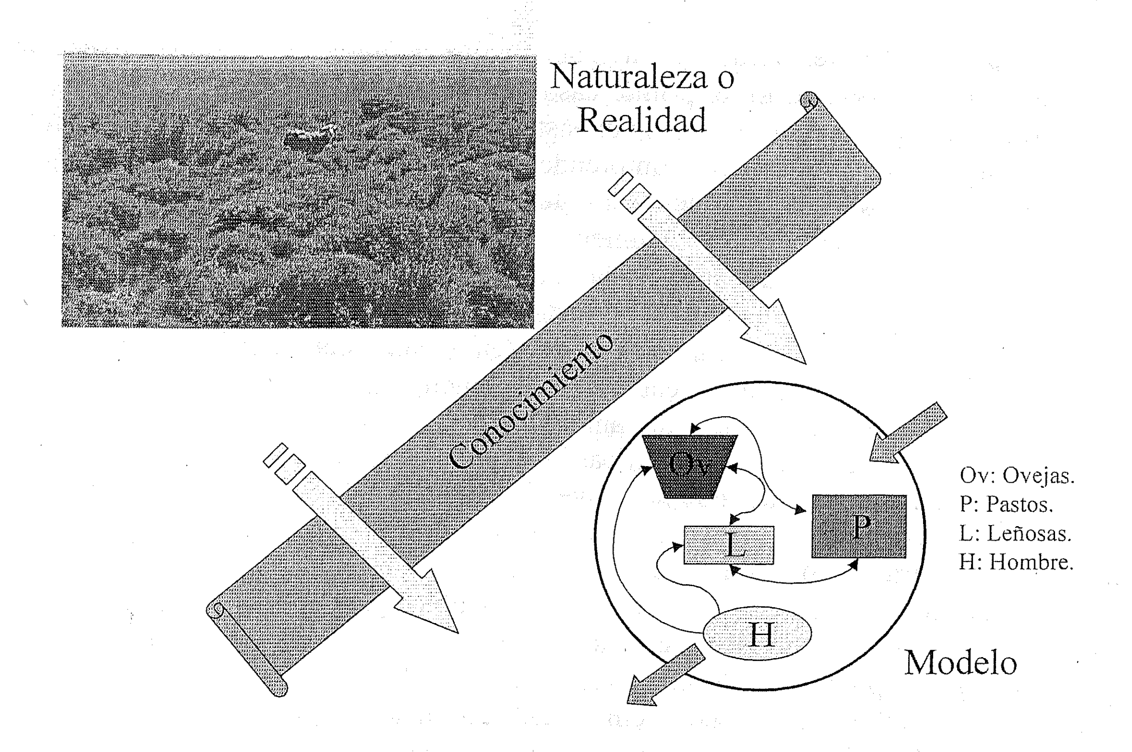



**Figura 1.** Esquema que representa la acción de modelar.

El lenguaje de programación de menor nivel de abstracción está representado por las instrucciones básicas que posee el microprocesador de la computadora. La programación se realiza mediante códigos binarios que tienen un correlato directo con las características electrónicas de los circuitos y las compuertas digitales. Los lenguajes de nivel de abstracción superior poseen instrucciones, reglas y métodos más cercanos al lenguaje y representaciones cognitivas humanas. Para que sea posible usar los lenguajes de más alto nivel, se requiere de otro programa (llamado compilador) que traduzca las instrucciones de alto nivel a instrucciones de bajo nivel de abstracción. Esta necesidad de traducción hace que los modelos programados en lenguajes de alto nivel funcionen por lo general más lentos que los de bajo nivel. En muchos casos esta reducción en la velocidad es imperceptible o aceptable. En cuyo caso, la elección de un lenguaje de alto nivel se justifica ampliamente debido a su mayor facilidad de uso y relación más directa al sistema a modelar. Durante el transcurso de este curso utilizaremos un lenguaje de alto nivel de abstracción. Así podremos concentrar nuestros esfuerzos en comprender los aspectos conceptuales de la construcción de modelos y realizar experimentos de simulación.

## ¿Por qué es útil modelar?

Los modelos son útiles porque nos ayudan a **comprender** y nos permiten **predecir** el comportamiento de un sistema. En el primer caso, un modelo es la formalización de parte o todo el conocimiento que se tiene acerca de un sistema (Figura 1). De este modo el modelado es una herramienta muy útil para comprender la complejidad de distintos sistemas, particularmente de los sistemas naturales (ecosistemas). Generalmente, la construcción de un modelo permite comprender el funcionamiento de entidades complicadas y pobremente conocidas a partir del comportamiento de las partes que son mejor conocidas (Goodman 1975). Los modelos entonces ayudan a comprender el funcionamiento de un sistema a través de la conceptualización y la organización de fenómenos complicados y facilitan la comunicación en forma explícita y relativamente sencilla del conocimiento referente al sistema. El ejercicio de modelado permite tanto generar y probar nuevas hipótesis acerca del funcionamiento de un sistema, como identificar los puntos donde la información es deficiente, con lo cual se convierte en una herramienta clave en el avance del conocimiento.

En el segundo caso, el papel de los modelos cobra una importancia extra en el contexto de la elaboración de pronósticos o evaluaciones a futuro (Clark et al. 2001). A medida que avanzamos en una era caracterizada por cambios globales, como son los cambios en el uso de la tierra, los cambios climáticos, el incremento de $CO_2$ atmosférico, las invasiones biológicas, problemas de contaminación, etc. (Vitousek et al. 1997a, b) resulta difícil predecir los cambios en los ecosistemas mediante metodologías tradicionales (Shugart y Post 1992, Wu y David 2002). Existe una carencia obvia de control experimental cuando cambios complejos ocurren a escala global, lo cual sumado a dificultades logísticas, confunden la evaluación de las consecuencias globales. Esto ha creado la necesidad de realizar extrapolaciones razonadas desde datos experimentales, observaciones y la aplicación de modelos para predecir la respuesta ecológica del globo a cambios en el ambiente (Shugart y Post 1992). Esta situación particular en la que se encuentran actualmente muchas ciencias, donde los estudios de largo plazo y la necesidad de efectuar pronósticos son imprescindibles para comprender la dinámica de los procesos y/o tomar decisiones, posiciona a los modelos en un lugar clave (Clark et al. 2001).

## El modelo como integración y generación de conocimiento

Como mencionamos en la sección anterior, uno de los principales beneficios que ofrecen los modelos es la posibilidad de resumir e integrar los procesos más importantes que rigen la dinámica de un sistema. Este ejercicio de integración permite detectar posibles áreas vacantes del conocimiento en las que se requiere profundizar para mejorar la comprensión global del sistema. Por otra parte, permite explorar situaciones aún no halladas en la realidad.

Por ejemplo, Austin y Vitousek (1998) y Austin (2002) estudiaron la dinámica de circulación de carbono (C), nitrógeno (N) y fósforo (P) en un gradiente de precipitaciones en Hawaii. Mediante distintos experimentos y mediciones en sitios con diferente precipitación anual (desde 500 hasta 5500 mm año$^{-1}$) analizaron la tasa de productividad y de descomposición de la vegetación, lo que les permitió caracterizar con buen nivel de detalle los flujos de C, N y P. Una de las conclusiones más importantes que obtuvieron fue que la cantidad de precipitaciones anuales del sitio era la variable más importante en controlar la dinámica de estos nutrientes. Para integrar los conocimientos adquiridos acerca de cada flujo particular diseñaron un modelo (Figura 2) que luego utilizaron para simular de manera conjunta los flujos de nutrientes.

En un experimento realizado con el modelo matemático, simularon el comportamiento del sistema durante 200 años (experimento imposible de concretar en el sistema real por investigadores mortales). De esta manera pudieron explorar el comportamiento del sistema en el estado de equilibrio[2]. Entre otras cosas encontraron que en el estado de equilibrio la cantidad de biomasa muerta era máxima en los sitios de 1500 mm año$^{-1}$, y que la concentración de N y P en la biomasa muerta era máxima en los sitios de 2000 mm año$^{-1}$. La integración, a través del modelo, de la información adquirida en diferentes experimentos, permitió detectar *propiedades emergentes* del sistema que no eran evidentes cuando cada flujo era analizado por separado. Como consecuencia, el modelo a su vez generó nuevo conocimiento.

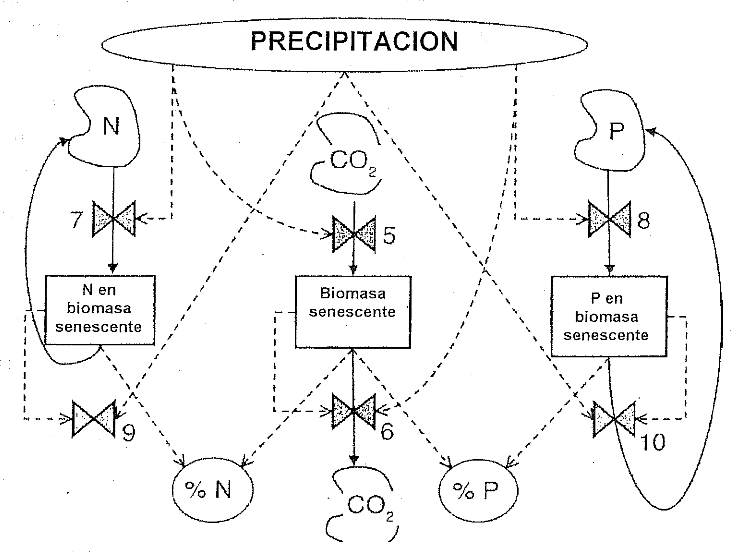


**Figura 2.** Diagrama esquemático de un modelo de simulación de la productividad, descomposición, concentración de nutrientes y liberación de nutrientes en función de la precipitación. Adaptado de Austin (2002).

## *Propiedades emergentes*

Las **propiedades emergentes** son cualidades de los sistemas jerárquicos, y los modelos han demostrado ser muy eficaces para su estudio. Los modelos permiten sintetizar el conocimiento actual en un marco teórico común y por lo tanto explorar las propiedades emergentes, es decir, las consecuencias de mecanismos conocidos o hipotéticos a mayores niveles de integración (Goudriaan et al. 1999, Peck 2004). Por ejemplo, supongamos que contamos con un conocimiento bastante ajustado del funcionamiento individual de cada uno de los componentes del sistema de la Figura 1. Aún así, difícilmente podamos predecir *a priori* el funcionamiento del sistema, en el cual los distintos mecanismos operan simultáneamente y los componentes interactúan entre sí. Es decir, que el conocimiento del comportamiento individual raramente es suficiente para predecir cual será el resultado de la integración de todos los componentes. Esto es así porque generalmente en los sistemas biológicos existe una jerarquía de dominancia de los mecanismos involucrados sobre la dinámica de un proceso y/o aparecen diferentes interacciones entre los mecanismos que hacen más complejas las respuestas y/o intervienen controles de autorregulación (ciclos de retroalimentación positiva y negativa). En consecuencia, el sistema en conjunto contiene más información que la suma de la información contenida en sus partes, ya que a partir de la relación entre las partes surge un conjunto no aditivo de propiedades emergentes.

## *El enfoque sistémico en la ciencia*

El análisis de los sistemas complejos como un todo y el modelado de esos sistemas son complementarios a las visiones *analítico - reduccionistas* en ciencia. Aislar y controlar componentes de la biosfera ha sido una de las maneras más empleadas para estudiar la naturaleza por parte del hombre. La corriente *reduccionista* es en cierta medida una ciencia de

[2] Estado de equilibrio de un sistema hace referencia a una condición particular del mismo donde todos los controles que influyen se cancelan, resultando en un estado estable, balanceado y sin cambios en el tiempo.

las partes que surge de la tradición de las ciencias experimentales, donde se elige un foco suficientemente estrecho para postular hipótesis, recolectar datos y diseñar pruebas apropiadas para refutar hipótesis no válidas. Debido a que se fundamenta en la experimentación, la escala elegida debe ser necesariamente pequeña en el espacio y en el tiempo. El objetivo de la ciencia de las partes es disminuir la incertidumbre hasta el punto donde la aceptación de un argumento es unánime. Es adecuadamente conservadora y no ambigua, pero a costa de ser fragmentaria y de escala reducida. Provee los ladrillos para el edificio pero no el diseño arquitectónico. En consecuencia, este enfoque por sí solo no permite comprender el funcionamiento del todo.

Los modelos han sido muy útiles en la generación y aplicación de conocimiento a través de visiones reduccionistas, pero han cobrado máxima importancia en la corriente holística. El *holismo* o enfoque de sistemas es la ciencia de la integración de las partes, que estudia el todo con sus componentes en forma interrelacionada (Figura 3). Utilizando también resultados de los estudios reduccionistas, identifica vacíos, desarrolla hipótesis alternativas causales, construye y utiliza modelos como recurso para la exploración y la experimentación. Evalúa la consecuencia de cada hipótesis alternativa utilizando información de intervenciones planeadas y no planeadas en el sistema completo o a través de la comparación y el contraste de ejemplos extremos. Las escalas de estudio elegidas son definidas por el problema en cuestión y no por limitaciones prácticas de la experimentación.

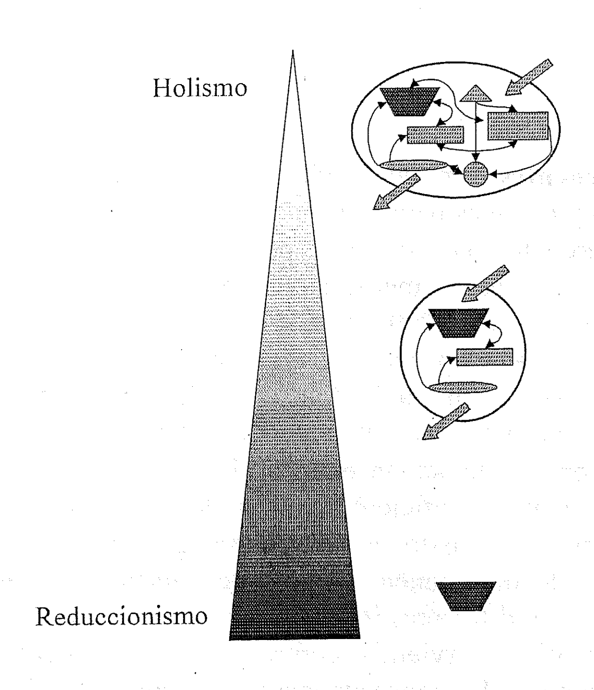



**Figura 3.** La pirámide esquematiza distintos niveles de complejidad y de percepción del funcionamiento de un componente en forma aislada o como parte del sistema.

## Modelos de amplia difusión en las ciencias del ambiente

Existen actualmente numerosos modelos que ayudan al hombre a abordar problemas de diversa naturaleza en la Tierra. Algunos modelos vinculados con las ciencias ambientales permiten predecir la dinámica tridimensional de un derrame de fluidos (ej. petróleo) en el agua (océano, ríos, etc. como MIKE11, HEC, QUAL2E, SBEACH entre otros), otros permiten predecir el incremento que sufrirá la temperatura media de la tierra en el largo plazo

si no se aplican políticas para regular las emisiones de gases invernadero, otros nos ayudan a predecir la dinámica de numerosos recursos naturales como pesqueros, madereros, etc. También existen muchos modelos que involucran procesos productivos, como aquellos que nos permiten conocer el rendimiento de diversos cultivos a partir de información eco-fisiológica (Oil-crop, Ceres, etc.). Otros modelos más tecnológicos nos permiten monitorear y mejorar procesos industriales. Estos modelos permiten, entre otras cosas, probar combinaciones de procesos, cronometrar sus tiempos e insumos utilizados, y sobre la base de los resultados mejorar la secuencia de tareas y/o el uso de insumos. A modo de ejemplo: algunas industrias productoras de cajas de cartón poseen modelos que diseñan los cortes de manera de minimizar el trabajo y el material desperdiciado. Otras industrias poseen modelos de optimización dedicados a elaborar logísticas de transporte o acopio: armar la ruta de distribución que minimice kilómetros recorridos u ordenar un depósito de artículos de manera de minimizar el tiempo y los movimientos para extraer o agregar nuevos artículos. En la Tabla 1, listamos a modo de ejemplo un conjunto limitado de algunos modelos muy difundidos en las ciencias agropecuarias y ambientales.

**Tabla 1.** Listado de algunos modelos muy conocidos y difundidos en las ciencias agropecuarias y ambientales.

| Nombre | Sistema |
|---:|---|
| Asko | Dinámica del ecosistema marino Báltico. |
| Barataray ecosystem | Dinámica de nutrientes en estuarios. |
| Biome-BGC | Ciclos del C, N y agua en ecosistemas. |
| Biospheric | Productividad del planeta Tierra. |
| Century | Materia orgánica del suelo. |
| Ceres | Cultivos. |
| Fire | Manejo del fuego en ecosistemas de bosque. |
| Formind | Bosques tropicales. |
| Genesys | Genética de poblaciones. |
| Jabowa | Dinámica de bosques. |
| Lotka-Volterra | Dinámica de poblaciones. |
| Oilcrop | Cultivos oleaginosos. |
| Patumod | Dinámica sucesional en pastizales alpinos. |
| Pest model | Dinámica de plagas y enfermedades de cultivos. |
| ConPast | Estimación del consumo de bovinos en pastoreo. |
| River | Flujo de ríos y productividad de estuarios. |
| Sortie | Bosques mixtos. |
| Steppe | Pastizales naturales. |
| Dinaqua | Dinámica del contenido hídrico del suelo en Patagonia |

Existen además muchos modelos utilizados por analistas económicos. Estos modelos permiten explorar escenarios futuros de una empresa, país o inclusive del mundo. Se usan habitualmente para realizar análisis de riesgo/beneficio. Varias de las técnicas empleadas por estos modelos son de utilidad para las ciencias ambientales y las veremos durante el curso.

## Clasificación de los modelos: Las distintas filosofías de modelado

*Modelos analíticos y de simulación:*

Este criterio de clasificación de los modelos hace referencia a la resolución matemática de sus algoritmos[3]. Los modelos analíticos presentan ecuaciones matemáticas complejas, su resolución se basa en métodos analíticos (e.g. sistema de pocas ecuaciones lineales diferenciales) y presentan mayores restricciones. La persona que modela a través de estos métodos tradicionalmente ha utilizado lápiz, papel y matemáticas complicadas. La aproximación analítica se refiere a un conjunto de procedimientos matemáticos para encontrar soluciones exactas a ecuaciones, entre otras, diferenciales. Ya que el comportamiento de cierta clase de ecuaciones matemáticas es bien conocido, la aproximación analítica tiene una gran capacidad predictiva en algunos aspectos de la naturaleza, siempre y cuando puedan ser descriptos con una o pocas de estas ecuaciones. Por ejemplo, esta aproximación ha sido exitosamente utilizada en física para estudiar interacciones entre algunas partículas. Mientras que las partículas respondan del mismo modo que las ecuaciones analíticas, el comportamiento de las partículas puede ser explorado para un amplio rango de condiciones, simplemente a través de su respuesta en las ecuaciones matemáticas. La aproximación analítica también ha sido utilizada en algunos aspectos de la ecología, especialmente en biología y genética de poblaciones. Sin embargo, ha sido de menor utilidad en el estudio íntegro de los ecosistemas. Esto se debe a que los modelos analíticos son útiles solamente cuando las ecuaciones que describen los procesos bióticos son lineales y/o cuando hay pocas ecuaciones a ser resueltas en el mismo momento. Los modelos de ecosistemas, en cambio, generalmente incluyen muchas ecuaciones a resolver de manera simultánea, que pueden ser lineales o no (Hall y Day 1975).

Los modelos de simulación (o numéricos) cuentan con reglas matemáticas más simples, no pueden ser resueltos por métodos analíticos (ya que pueden comprender sistemas de muchas ecuaciones no lineales), sino más bien por métodos numéricos iterativos y presentan menos restricciones. El modelado de simulación está normalmente caracterizado por el uso de computadoras. Cuando consideramos procesos continuos, a diferencia de los modelos analíticos, los modelos de simulación no dan soluciones exactas. Sin embargo poseen varias ventajas, entre ellas, se pueden resolver varias ecuaciones de manera simultánea y es posible incluir ecuaciones no lineales. En muchos casos, esto permite la construcción de modelos que representen de manera más adecuada la realidad, sin la necesidad de suposiciones simplistas.

Ejemplos de aplicaciones muy útiles con ambos tipos de modelos coexisten en las ciencias naturales y ambientales. A pesar de ello, actualmente han cobrado mayor uso los modelos de simulación, especialmente en aproximaciones holísticas. Esto se debe a su mayor capacidad para abordar problemas complejos (e.g. se puede modelar de manera conjunta procesos interactivos entre múltiples componentes de un sistema), y a la posibilidad de abordarlos mediante el uso sencillo de matemáticas en conjunto con computadoras. Es por ello que, en este curso, nos focalizaremos en el estudio y en la comprensión de los modelos de simulación.

[3] Una lista de instrucciones lógicamente ordenadas, secuenciales y finitas donde se especifica una sucesión de operaciones necesarias para resolver algún problema particular.

*Otras clasificaciones:*

Además de las clasificaciones de modelos en físicos o digitales, analíticos o de simulación vistas anteriormente, existen múltiples criterios para clasificar los modelos. Un primer criterio hace referencia al modo en que representamos un modelo. En este sentido un modelo puede ser representado por conceptos, por diagramas o gráficos, por ecuaciones matemáticas o directamente por líneas de código en algún lenguaje de programación.

Otro criterio de clasificación hace referencia a la causalidad de las relaciones funcionales incorporadas en un modelo. Podemos hablar de modelos descriptivos cuando comprenden relaciones entre variables que no necesariamente involucran mecanismos causa-efecto. O hablar de modelos mecanísticos cuando las relaciones incorporadas se basan en un profundo conocimiento de las relaciones causales.

Los modelos también pueden ser clasificados según la posibilidad de incorporar variación temporal aleatoria en sus parámetros. En este sentido hablamos de modelos determinísticos o modelos estocásticos. Los primeros se refieren a modelos en los cuales los parámetros representan constantes desde el punto de vista matemático y no varían durante su resolución. Por lo tanto, el resultado proveniente de simulaciones con modelos determinísticos se mantiene constante para un dado conjunto de parámetros. En cambio los modelos estocásticos, en vez de incorporar un único valor del parámetro, incorporan una distribución de probabilidad. Por ejemplo, un parámetro X puede ser definido como una distribución Normal con media $\Pi_x$ y desvío $\varsigma_x$. De este modo, el valor de un parámetro del modelo puede variar acorde con su distribución durante la resolución o realización del mismo y por ende el resultado del modelo no se mantendrá constante entre simulaciones.

Otro criterio de clasificación de los modelos hace referencia a su capacidad para representar el espacio físico. Los modelos tradicionales sólo reflejan cambios mayormente en el tiempo sin tener en cuenta la dimensión espacial, mientras que aquellos que si lo hacen son conocidos como modelos espacialmente explícitos. Finalmente, un criterio de clasificación relativamente actual hace referencia a la idea de modelos basados en individuos o modelos que no conservan tal identidad. A modo de ejemplo uno puede simular la dinámica de una población o comunidad a partir de modelar el ciclo de vida de distintos individuos y sus interacciones con parámetros que definen el comportamiento de cada parte (individuo), o modelar el comportamiento del todo. Durante el curso veremos, por ejemplo, un modelo de simulación de la relación depredador-presa que corresponde al primer tipo (simula nacimientos y muertes de individuos en las poblaciones) y otro de dinámica del agua en un pastizal que corresponde al segundo.

## ¿Cómo modelamos?

Modelar requiere primero un objetivo, una pregunta o un problema de interés para ser abordado y además suficiente conocimiento previo del sistema a modelar. Este es el primer paso para construir un modelo ya que permitirá definir claramente los límites, restricciones y supuestos sobre los que descansa el modelo (Figura 4). En una primera etapa, se incluirán de un modo conceptual o esquemático las partes que se consideran relevantes al sistema a modelar y la forma en que ellas interactúan. Mucho del logro durante esta etapa depende del conocimiento que se tenga del sistema.

La siguiente etapa requiere de un conocimiento más profundo de las relaciones entre los componentes del modelo ya que se necesita formalizar en términos de algoritmos matemáticos el comportamiento de cada parte y su interacción con otros componentes. Luego es necesario codificar esta información en un ambiente adecuado que permita resolver las relaciones incluidas en el modelo y observar el comportamiento del mismo.

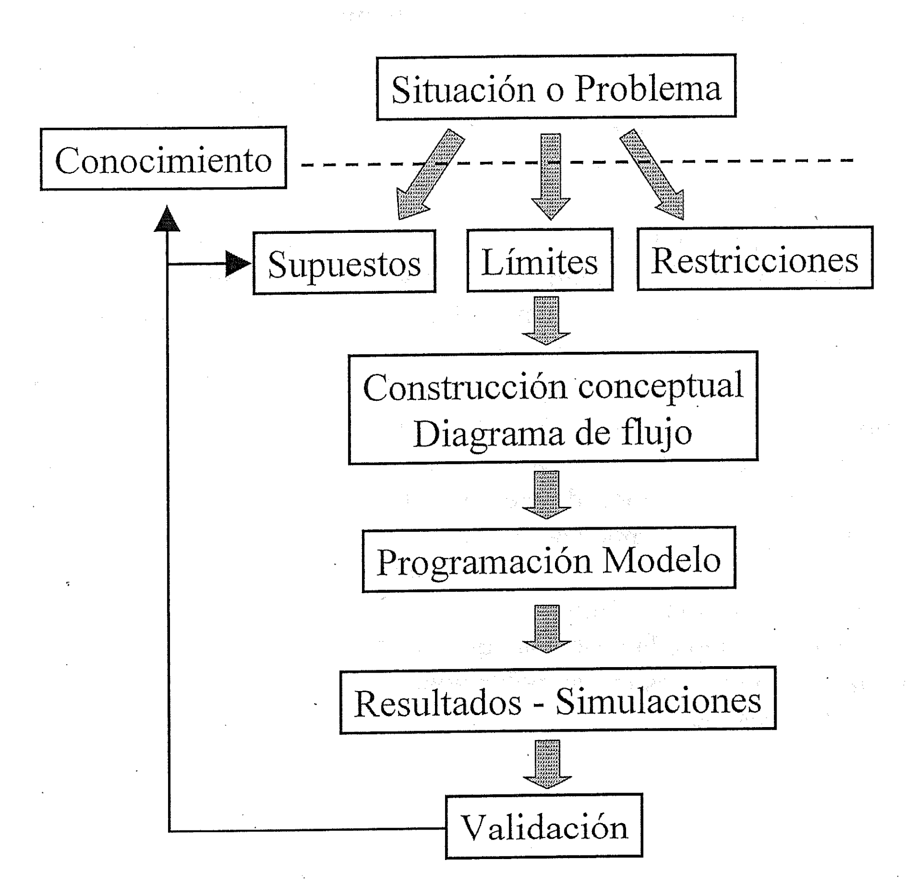



**Figura 4.** Cuadro esquemático con las diferentes etapas involucradas en la construcción y evaluación de un modelo.

La etapa final en la construcción del modelo es la validación, donde los resultados producidos por el modelo son contrastados contra situaciones equivalentes en la realidad para medir el grado de éxito o poder predictivo del mismo. Si la validación no es satisfactoria habrá que revisar nuevamente todo el modelo desde sus supuestos hasta sus relaciones. En cambio si esta etapa es exitosa se puede comenzar a estudiar el comportamiento del sistema, generar y probar nuevas hipótesis para ampliar el conocimiento del sistema. Eventualmente, se puede incorporar *a posteriori* dentro del modelo el nuevo conocimiento generado.

A continuación ahondaremos detalladamente en dos modelos que utilizaremos en el transcurso de los trabajos prácticos. En principio explicaremos un modelo sencillo de redes

circulares desarrollado por el Ing.Agr. Gustavo Sznaider, luego continuaremos con un modelo simple de balance hídrico implementado en Excel desarrollado por el Ing. Agr. Gonzalo Grigera y finalmente estudiaremos las relaciones entre hierbas, pastos, leñosas y ganado vacuno en un modelo complejo para simular la dinámica de pasturas desarrollado por el Dr. Francois Gillet para los valles franceses.

### *Modelo de redes circulares*

Los modelos piedra, papel y tijera (un tipo particular de red circular) se utilizan para estudiar sistemas compuestos por tres poblaciones que compiten entre sí. En estos sistemas ocurre una relación de dominancia circular, donde individuos de la 1era población triunfan sobre individuos de la 2da, individuos de la 2da triunfan sobre individuos de la 3era e individuos de la 3era triunfan sobre individuos de la 1era. A este tipo de relación se la denomina "piedra, papel, tijera" por su similitud con el juego homólogo.

Las relaciones de dominancia cíclica han sido observadas en la naturaleza. Existe por ejemplo 3 tipos de Escherichia Coli, una que produce una toxina, otra que es susceptible y otra que es resistente. Cuando se juntan la resistente con la susceptible está última triunfa puesto que no destina energía a poseer resistencia y destina mayor cantidad de energía a las funciones habituales de reproducción y metabolismo de alimento. La productora de toxina obviamente aniquila a la susceptible. Cuando se juntan la productora de toxina con la resistente, está última triunfa en la competencia puesto que la energía necesaria para producir la toxina es superior a la necesaria para poseer resistencia (Kerr et al, 2002). Otros ejemplos son la relación entre 3 especies de abejas en Japón (Tainaka y Nakagiri, 2000) o la existente entre 3 variedades de lagartijas (Sinervo y Lively 1996).

El estudio de la dominancia circular no se restringe a sistemas de 3 especies. Durante los últimos tiempos se han estudiado sistemas de más de 3 especies (Szabó y Sznaider 2004) donde además de la relación circular existen otros tipos de relaciones (Szabó y Czárán, 2002). Por lo general estos tipos de sistemas muestran comportamientos interesantes y características que sirven para elaborar hipótesis sobre mecanismos de coexistencia y biodiversidad.

### *Modelo de balance hídrico*

El objetivo de este modelo de balance hídrico es calcular diariamente la evapotranspiración y el contenido de agua útil en el suelo (CAU) de pasturas y pastizales. La evapotranspiración es el flujo de agua en fase vapor que se pierde hacia la atmósfera desde una superficie vegetada, y resulta de la suma de la evaporación directa desde el suelo y la transpiración de las plantas, principalmente a través de los estomas. El CAU de un suelo representa la fracción del agua contenida que un tipo de vegetación determinado es capaz de absorber, y se calcula como la diferencia entre el contenido de agua a capacidad de campo (CC) y en el punto de marchitez permanente (PMP). Las variables de salida calculadas son luego utilizadas en otro modelo que estima la productividad forrajera, por lo que el modelo de balance hídrico puede ser visto como un submodelo o un módulo de un modelo más general. Su estructura es sencilla, ya que solo tiene una variable de estado: el contenido de agua útil en el suelo (CAU, en mm); y tres variables de flujo: la precipitación (Ppt), la evapotranspiración actual (ETA) y la percolación profunda (Perc; en mm/día; Figura 5).

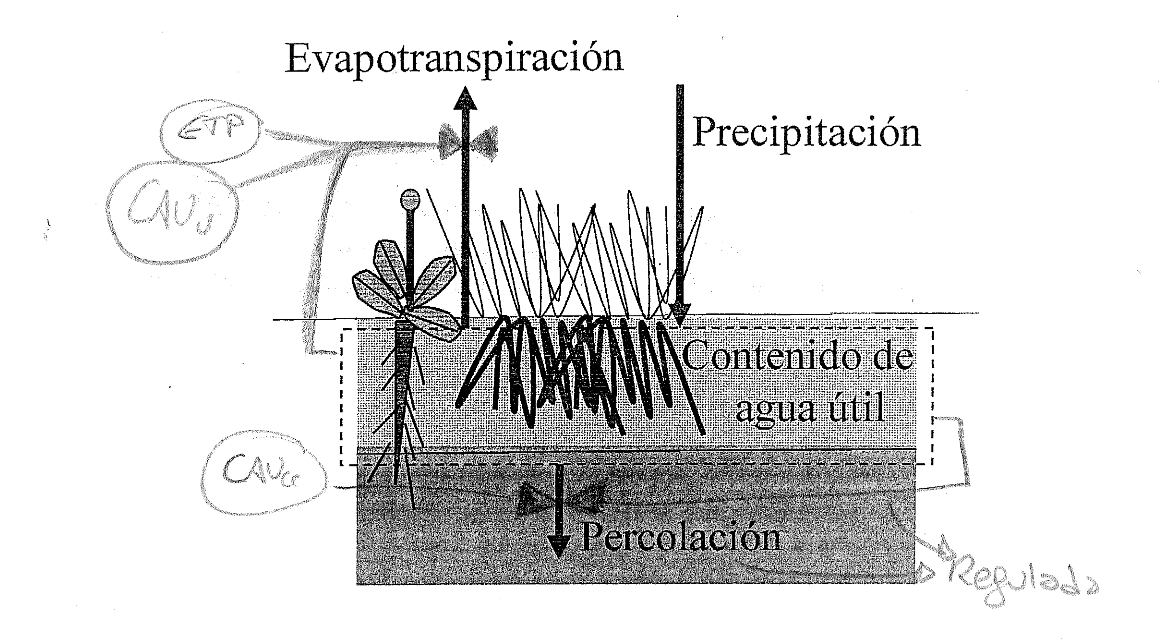



**Figura 5.** Representación del modelo de balance hídrico. La variable de estado está representada por la caja punteada y las de flujo por las flechas.

En el modelo, el CAU en el día *i* depende del CAU en el día anterior y de la Ppt, la ETA y la Perc de ese día:

$$CAU_i = CAU_{i-1} + \Delta CAU$$

$$\frac{\Delta CAU}{\Delta t} = Ppt_i - ETA_i - Perc_i$$

La ETA en el día *i* depende de la evapotranspiración potencial (ETP) de ese día y del CAU en el día anterior, siempre que dicho CAU resulte menor a un valor umbral. En caso contrario, la $ETA_i$ es igual a la $ETP_i$:

$$ETA_i = ETP_i * \frac{CAU_{i-1}}{CAU_u}, \qquad \text{si } CAU_{i-1} < CAU_u;$$

$$ETA_i = ETP_i, \qquad \text{si } CAU_{i-1} \geq CAU_u,$$

donde CAUu es un contenido de agua útil umbral por debajo del cual la vegetación se supone bajo estrés hídrico por lo que no transpira a la tasa potencial (Figura 6). El CAUu es definido como una proporción del contenido de agua útil a capacidad de campo (CAUcc) e idealmente es determinado en condiciones experimentales. La ETP diaria es calculada según el método de Priestley-Taylor, que se basa en datos de radiación y de temperatura.

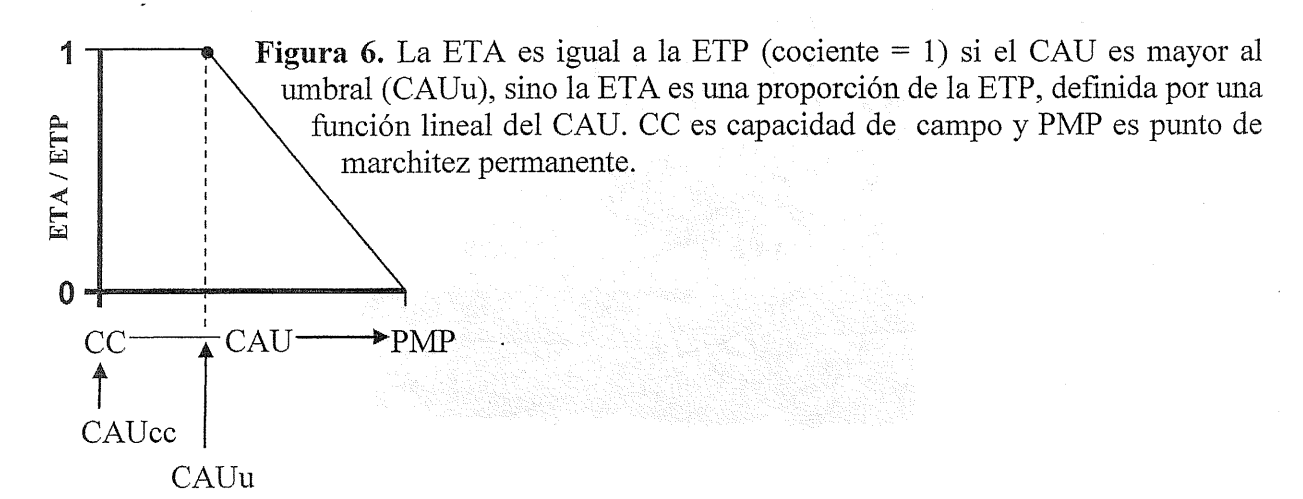



**Figura 6.** La ETA es igual a la ETP (cociente = 1) si el CAU es mayor al umbral (CAUu), sino la ETA es una proporción de la ETP, definida por una función lineal del CAU. CC es capacidad de campo y PMP es punto de marchitez permanente.

Se asume que la percolación es el excedente de agua sobre la cantidad que ese suelo en particular es capaz de retener:

$$Perc_i = \frac{\max(CAU_{i-1} + Ppt_i * \Delta t - ETA_i * \Delta t - CAU_{cc};0)}{\Delta t}$$

El cociente ETA/ETP en un día determinado, es un indicador de la condición hídrica del cultivo porque informa qué proporción del agua que se habría evapotranspirado en una situación de agua no limitante efectivamente se evapotranspiró. Por ello, esta variable de salida del modelo es luego utilizada en el modelo de productividad forrajera como un control de la productividad. En la Figura 7 pueden observarse los resultados de una simulación, que incluyen al cociente ETA/ETP, para una pastura de Coronel Suárez en el ciclo de producción 2002-2003. Puede notarse claramente el comienzo de la intensa sequía que afectó al sudoeste de la provincia de Buenos Aires durante todo el año 2003.

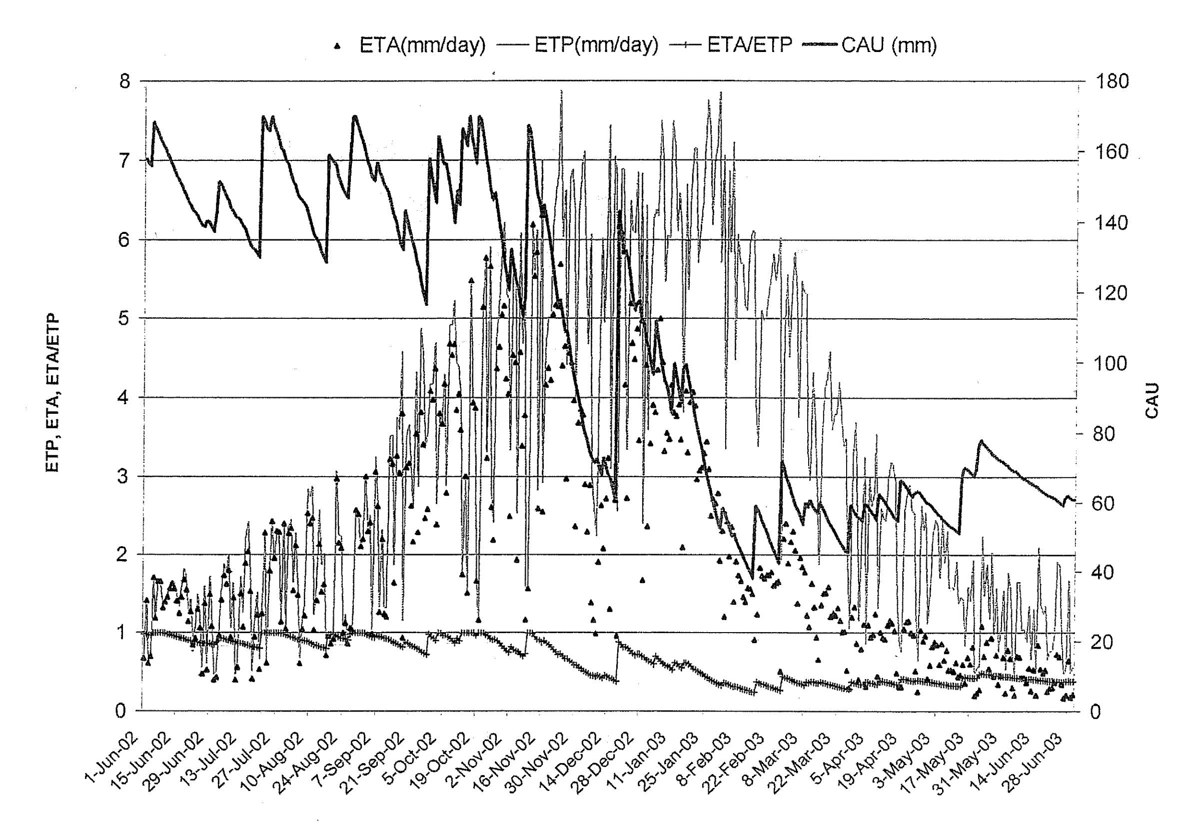



**Figura 7.** Salidas de una simulación del balance hídrico de una pastura de Coronel Suárez. El cociente ETA/ETP es un indicador de la condición hídrica a la que se encuentra expuesta la pastura.

## *Modelo de dinámica de pasturas*

El modelo de dinámica de pasturas (PATUMOD) simula y predice la sucesión de la vegetación de una pastura con componentes leñosos (árboles y arbustos) y herbáceos. La simulación se realiza a partir de condiciones iniciales, bajo la influencia de determinadas condiciones abióticas y del manejo del pastoreo a través del control de la carga animal y la acción antrópica sobre los bosques (raleo). El modelo se halla compuesto por un conjunto de comunidades vegetales vinculadas por una red de relaciones temporales. La estructura de dicha red es la encargada de definir la dinámica temporal del sistema. Las principales variables de ingreso al modelo definen las condiciones iniciales de cobertura de cada tipo de comunidad vegetal (cobertura de mallines, arbustales, bosques, etc.), un conjunto de variables ambientales del modelo (ej. altitud, carga animal, valor forrajero de cada comunidad vegetal, etc.) y finalmente un conjunto de parámetros que definen las actividades antrópicas (tasas e intervalos de raleo, etc.). La principal variable de respuesta (salida) del modelo es la cobertura de las distintas comunidades vegetales.

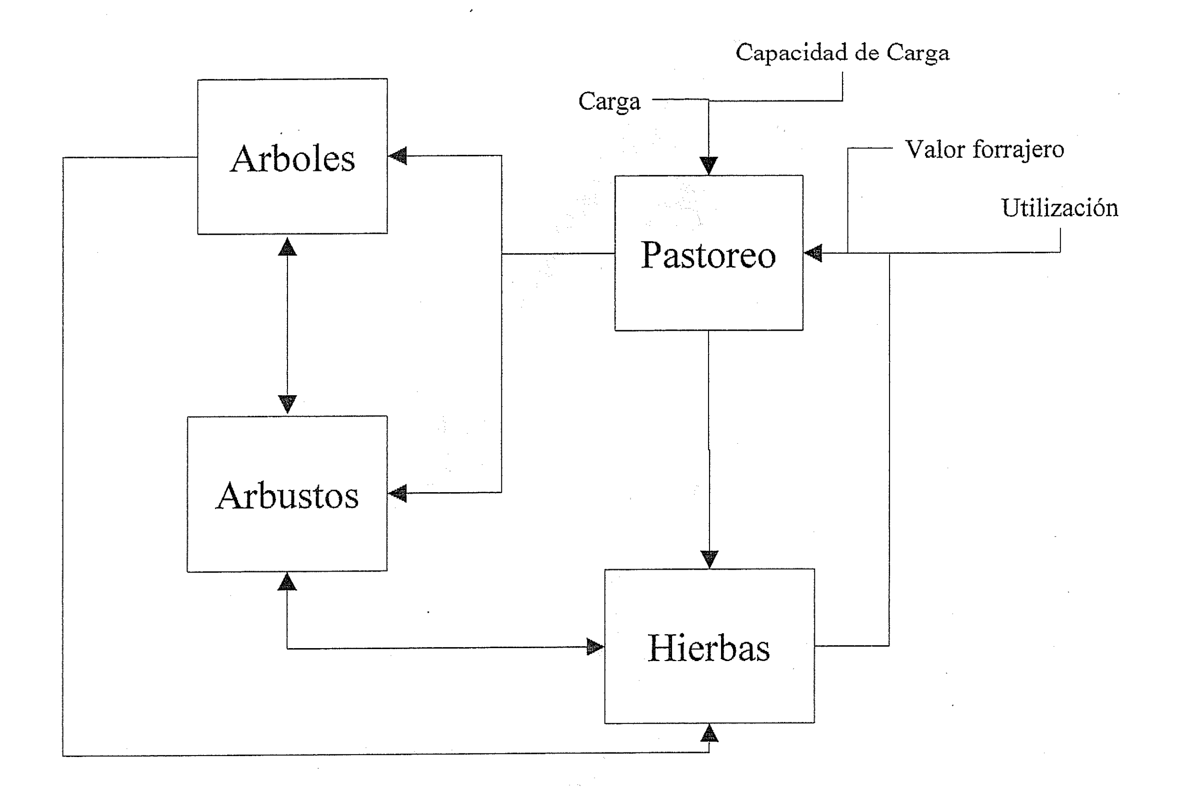



**Figura 8.** Representación simplificada del modelo de dinámica de pasturas (Patumod). Los principales sectores del modelo están representados por cajas y sus interconexiones a través de flechas. En forma detallada se muestran los controles para el módulo de pastoreo.

# Algoritmos

Todos los modelos necesitan especificar las reglas de su funcionamiento. Estas reglas de funcionamiento pueden variar desde simples definiciones coloquiales hasta complejas ecuaciones matemáticas. En general hablamos entonces de *algoritmos* haciendo referencia a **una lista de instrucciones lógicamente ordenadas, secuenciales y finitas donde se especifica una sucesión de operaciones necesarias para resolver algún problema particular**. Generalmente la mayoría de los modelos presentan varios algoritmos que permiten resolver pequeñas partes del problema en cuestión para luego ser integradas y dar respuesta al problema general. En el ejemplo del modelo de balance hídrico podemos ver a un algoritmo encargado de resolver la estimación de la evapotranspiración y otro para la percolación, sin embargo estos representan pasos intermedios para la obtención del contenido de agua útil del suelo.

Los algoritmos y el análisis de los mismos constituyen una parte importante del *procesamiento de información*, que permite a partir de una información primaria o de entrada (input) obtener un resultado depurado con algún grado de procesamiento (output) (Figura 9). Para esto, los distintos elementos que generalmente están incorporados en la mayoría de los algoritmos son: datos en forma de variables y / o constantes, controladores de flujo, condicionales y funciones matemáticas.

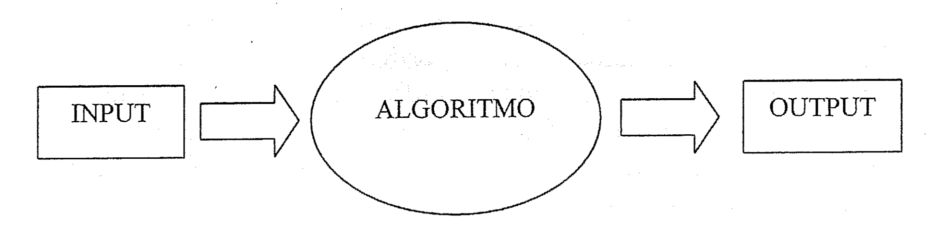



**Figura 9.** Esquema básico del funcionamiento de un algoritmo.

La *programación* es una aproximación metodológica que nos permite volcar la información conocida o supuesta acerca de un sistema real en un sistema virtual representado en algún tipo de código (e.g. compiladores, lenguajes o ambientes de programación) con el fin de **resolver problemas**. Por esto, programar es más que saber escribir una serie de instrucciones en algún código o lenguaje de programación (e.g. Basic, Pascal, Delphi, Squeak, Stella, Fortran). En este sentido, el **primer paso** en la programación sería conocer y diseñar conceptualmente y / o gráficamente el problema en cuestión, para luego avanzar en la sintaxis propiamente dicha. Parte de esta primera etapa debería consistir en identificar y describir muy bien el problema, saber por qué es necesario resolverlo, imaginar que resultados esperaría obtener y si es posible desarmar el problema en partes más pequeñas dentro de alguna jerarquía. Como **segundo paso**, debemos decidir cual es la información que queremos obtener y que nos permitirá resolver nuestro problema. En esta etapa es bueno pensar no sólo en el contenido de la información en si mismo, si no también en su presentación, es decir si un gráfico, una tabla, una impresión en pantalla o en papel, un listado almacenado en un archivo son algunas de las distintas maneras de presentar la información. Como **tercer paso**, necesitaríamos decidir y preguntarnos acerca de que datos debemos procesar con el programa para alcanzar la información que resuelve nuestro problema definido en el paso anterior. En otras palabras, cuales serán los datos crudos que permiten a través de algún procesamiento, arribar a la información de interés. Finalmente, como **último paso**, es útil volcar todo esto en

un esquema o plan (pseudo-código) que guíe el traspaso de lo que queremos programar a algún lenguaje que nos sirva de intérprete entre nosotros y la computadora.

Para todo esto, nos valemos de nuestro *conocimiento* del problema y de las *computadoras* que son herramientas muy útiles para manejar datos, declarar variables, objetos, rutinas, sub-rutinas o funciones, diseñar la lógica de ejecución de los algoritmos, etc. que nos permiten procesar los *datos* de acuerdo con las necesidades del problema para obtener *información* que facilite la toma de decisiones. Entonces, hablamos de datos cuando hacemos referencia a la unidad básica a partir de la cual derivamos la información a través de algún procesamiento. Luego de haber diagramado el problema, el mismo es volcado en algún lenguaje o ambiente de programación, para luego inmediatamente verificar mediante su uso si funciona (si corre). Lo más habitual es que existan errores de programación, es decir problemas que tienen que ver con la sintaxis o un mal manejo del ambiente de programación. Por suerte, los compiladores actuales tienen numerosas herramientas que nos permiten revisar el programa mientras se ejecuta y detectar y corregir los errores (e.g. una inconsistencia en la declaración de una variable). Esta práctica se conoce en la jerga como "debugging" que traducido al español sería algo así como "eliminar los bichos" donde "bichos" hace alusión a los errores. Una vez corregidos los errores de programación el programa se ejecuta, pero aún puede tener errores, sobretodo de los más importantes referentes a la conceptualización del problema. Para esto durante esta etapa, es muy importante verificar si los resultados arrojados por el modelo son coherentes con lo esperado en el sistema real. Finalmente, una vez que el programa corre y se ejecutan correctamente sus algoritmos, y se alcanza plena certeza acerca de que está haciendo lo correcto y existe plena conformidad con los resultados observados ante parametrizaciones distintas contrastadas contra el modelo real, se puede ingresar en una última etapa con el fin de agilizar el procesado del algoritmo. Esta última etapa también es muy importante y puede ser clave cuando el tiempo de procesamiento es una variable a tener en cuenta. Por ejemplo, este aspecto es muy importante en problemas de optimización, dónde hay que evaluar una cantidad muy grande de combinaciones de niveles de distintos factores y las dimensiones del problema pueden hacer que explorarlo de uno u otro modo, haga que la situación sea viable o imposible.

En la vida diaria cuando uno de nosotros quiere expresar un pensamiento, opinión o compartir algo con otro lo hacemos a través de alguna *expresión.* Existen diversos tipos de expresiones, algunas de las más comunes pueden ser gestuales, escritas, u orales. Entonces si emisor y receptor comprenden el mismo código el mensaje será transmitido exitosamente. En programación, cuando uno quiere transmitir a la computadora alguna operación (e.g. suma, regla lógica, etc.), esta también se hace a través de expresiones que se conocen como instrucciones y que generalmente permiten construir los algoritmos. El programador se vale entonces de distintos tipos de expresiones para declarar variables, asignar datos a variables, controlar el flujo de ejecución de un programa, almacenar resultados, establecer reglas lógicas, etc. A continuación vemos algunas de las expresiones más empleadas en programación.

Un aspecto inicial en programación es la definición del *tipo de datos* y la declaración de las *variables* empleadas en el algoritmo. Pero, ¿qué es una variable? Rápidamente definiremos variable como **un atributo que contiene datos a los fines del problema entre manos**. Una definición más técnica de variable podría ser una posición o lugar en la memoria de la computadora destinada a almacenar datos. Por ejemplo, en el modelo de balance hídrico, la precipitación diaria representaba una variable. Una primera clasificación de las variables determina **variables numéricas y de texto** (o cadena). Además, dentro de esta clasificación,

las variables numéricas pueden definirse de acuerdo a si son enteras o decimales y al grado de precisión de las mismas. También los algoritmos suelen contener *constantes,* que representan a aquellos datos que nunca varían a lo largo de la ejecución del algoritmo. Finalmente, pueden existir variables más complejas conocidas como estructuras o ordenamientos en forma vectorial o matricial de los datos, en estos casos se suele hablar de *arrays* y estos permiten ordenar los datos como vectores o matrices de *n* dimensiones de acuerdo a criterios definidos por el programador. Por ejemplo, en el modelo de balance hídrico, se podría definir una variable vectorial para la precipitación con 365 elementos que almacenen la precipitación de cada día para un año en particular.

Los *controladores de flujo* **administran el orden de ejecución de las diferentes sentencias incorporadas en el algoritmo**. Los controladores de flujo junto con los condicionales, permiten en gran medida manejar la lógica de ejecución dentro del algoritmo. Existen algunas diferencias en la sintaxis de los distintos lenguajes de programación más utilizados (Basic, Delphi, Fortran, C, etc.) en lo que respecta a la programación de los controladores. Sin embargo, todos permiten por ejemplo repetir un conjunto de instrucciones *n* veces (bucle), o repetirlo hasta que suceda algo o moverse líneas arriba o abajo del código. Generalmente los controladores se hallan constituidos por un comando, luego una serie de instrucciones contenidas en el bucle de iteración, y finalmente un comando de finalización de la iteración que puede ser opcional. A continuación se indica un ejemplo de un bucle sencillo programado en C con el comando *for*,

```
for (int k=0; k<z; k++)
        {
         tamaño [k]++;
        }
```

allí se define el contador *k* como una variable entera (int) inicializada en 0 que presentará un incremento unitario por ciclo (*k*++) hasta alcanzar el valor de *z*. Siempre que *k* sea menor a *z* el fragmento de código entre llaves será leído y ejecutado. En este caso, cada vez que el bucle se ejecute, se sumará una unidad al *k-ésimo* elemento del vector *tamaño*.

Los *condicionales* actúan como filtros, básicamente **restringen la libre circulación de la ejecución del programa en función de la restricción lógica establecida** en el condicional. Existen condicionales simples o complejos, Al igual que los controladores, los condicionales se hallan conformados por una serie de comandos, la definición de la condición o restricción impuesta, un conjunto de instrucciones a ejecutar si el condicional arroja uno u otro resultado y un comando de finalización del condicional. A continuación se muestra un ejemplo de un condicional programa en C con los comandos *if* y *else if*,

```
if ( (bio>0) && (bio<(maxarbbio/3))) fps=1;
else if ((bio>=(maxarbbio/3)) && (bio<(maxarbbio*2/3))) fps= 0.07 +(0.0006*bio));
else if (bio>=(maxarbbio*2/3)) fps=2;
```

en la primera línea del algoritmo se establece la primera regla lógica con el comando *if*, que si la misma luego de ser evaluada es verdadera asigna el valor de 1 a la variable *fps*. Para esto la variable *bio* debe ser mayor a cero y menor a 1/3 de la variable *maxarbbio*. Si en cambio, la primera regla lógica es falsa se evalúa entonces la segunda que establece que si la variable *bio* toma valores intermedios entre 1/3 y 2/3 de la variable *maxarbbio*, la variable *fps* se relaciona linealmente con la variable *bio* de acuerdo con la función 0.07+0.0006**bio*. Finalmente, si la segunda regla lógica también es falsa y entonces la variable *bio* es mayor o igual a 2/3 de la variable *maxarbbio*, entonces *fps* vale dos.

Las *funciones y ecuaciones matemáticas* son muy utilizadas en programación. Generalmente se usan para **relacionar la información de entrada con la información de salida y son el núcleo** de los *algoritmos* (ver Figura 9). En términos del problema, permiten representar los distintos procesos simulados o simplemente analizar la información. Por ejemplo, se podría construir un programa muy sencillo para calcular la media muestral de un conjunto de datos. En este caso la función matemática sería,

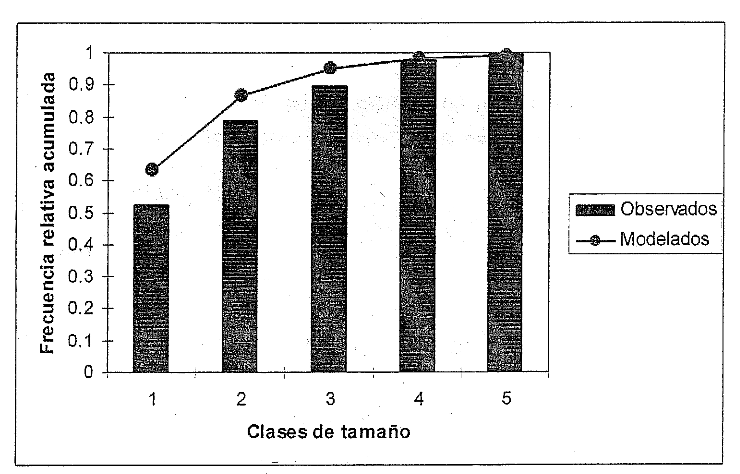
$$f(x) = \bar{x} = \frac{\sum_{i=1}^{i=n} x_i}{n}$$

y permitiría resumir la información de entrada con un estadístico descriptivo. En otros casos las funciones matemáticas son empleadas para simular un proceso o una situación observada en la naturaleza.

En este otro ejemplo, la distribución acumulada relativa de tamaños de árboles (DAP[3]) de un bosque se ajusta a una función de distribución de Weibull,

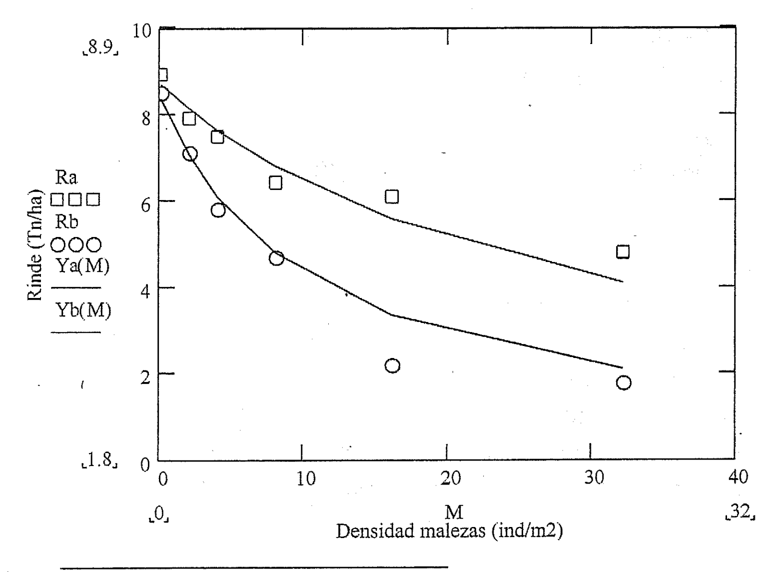
| DAP (cm) | Clase de tamaño ($c$) | Número de individuos |
|---|---|---|
| <4 | 1 | 103 |
| 4-8 | 2 | 52 |
| 8-12 | 3 | 21 |
| 12-16 | 4 | 16 |
| 16-20 | 5 | 4 |

$$F(c) = 1 - e^{-\left(\frac{c}{\alpha}\right)^{\beta}}$$




donde (c) representa las distintas clases de tamaños y alfa y beta los parámetros del modelo, que son en este caso iguales a uno.

En el siguiente ejemplo, se muestra la caída en el rendimiento de dos cultivares de maíz en función de la densidad de malezas y su respectivos ajustes a funciones hiperbólicas.

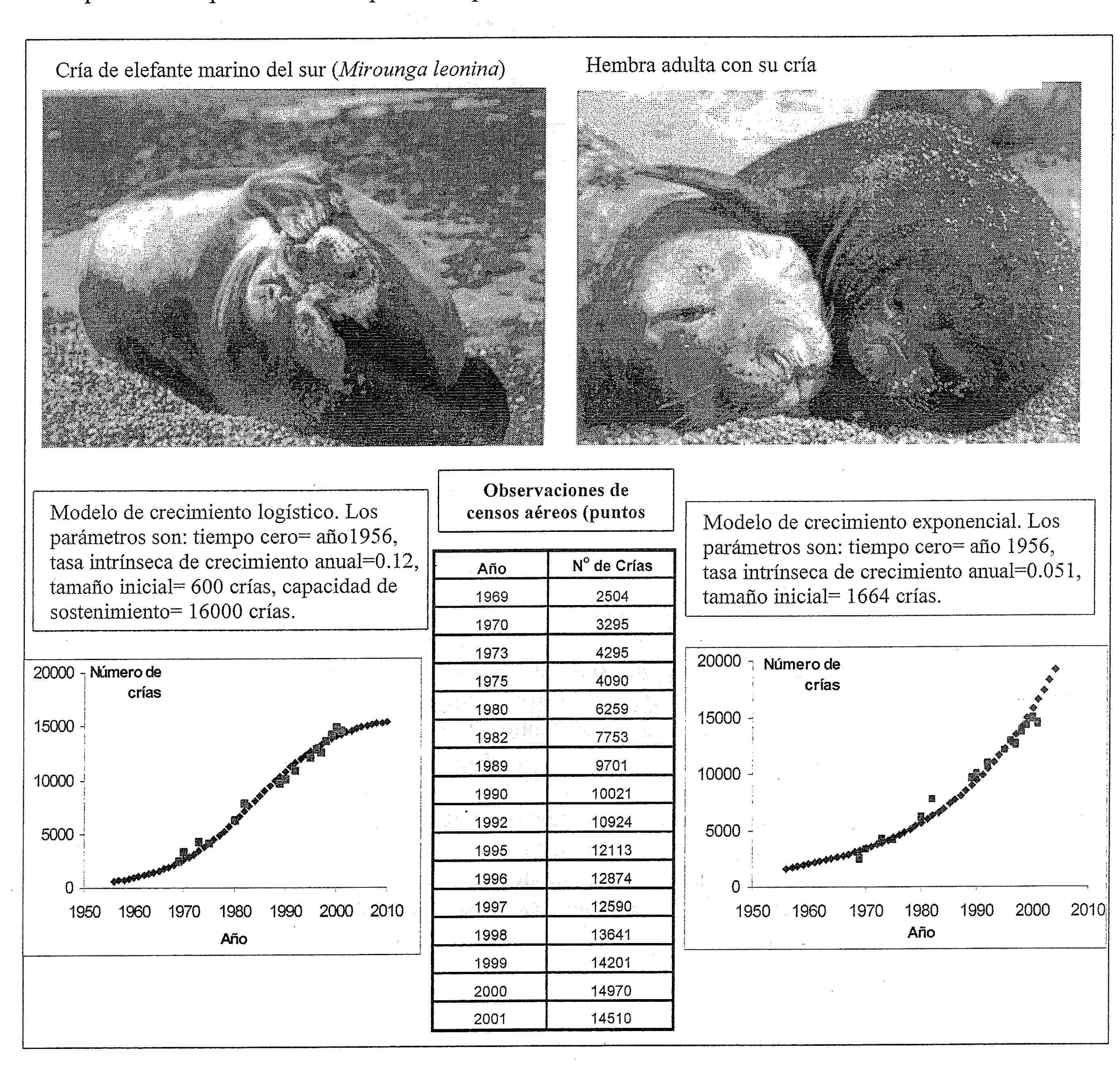



$$Ya(M) = 8.9\frac{1-0.026}{1+\frac{0.026M}{0.75}}$$

$$Yb(M) = 8.83\frac{1-0.054}{1+\frac{0.054M}{0.585}}$$

[3] Diámetro a la altura del pecho (1.4 m).

Este último ejemplo representa la dinámica de las poblaciones de elefantes marinos en la "élefantería" de Península de Valdés. Para esto, la información de los censos de número de crías, realizados por investigadores entre 1969 y 2001, se volcaron en gráficos que representen el tamaño poblacional y la tasa de crecimiento poblacional, que luego pueden compararse con los correspondientes a curvas de crecimiento exponencial y logístico (Figura 10). En la Tabla 2 se resumen a modo ilustrativo sólo algunas de las funciones matemáticas más empleadas en problemas ecológicos y ambientales, mientras que en la Tabla 3 se resumen los pasos conceptuales más importantes para escribir una ecuación.




| Año | N° de Crías |
|---|---|
| 1969 | 2504 |
| 1970 | 3295 |
| 1973 | 4295 |
| 1975 | 4090 |
| 1980 | 6259 |
| 1982 | 7753 |
| 1989 | 9701 |
| 1990 | 10021 |
| 1992 | 10924 |
| 1995 | 12113 |
| 1996 | 12874 |
| 1997 | 12590 |
| 1998 | 13641 |
| 1999 | 14201 |
| 2000 | 14970 |
| 2001 | 14510 |

**Figura 10.** Modelos de crecimiento poblacional ajustados a los datos de la "elefantería" de Península Valdés (fotos y datos cortesía Dra. Mirtha Lewis). Los puntos coloreados en los dos gráficos corresponden a las observaciones de censos aéreos que se presentan en la tabla central.

**Tabla 2.** Algunos ejemplos de funciones matemáticas y estadísticas muy aplicadas en distintos problemas agronómicos, ecológicos y ambientales.

| Función | Denominación | Aplicaciones |
|---|---|---|
| $Y = \alpha + \beta e^{\delta x}$ | Exponenciales | Dispersión de semillas |
| $\frac{\partial y}{\partial t} = 32 - \frac{\alpha y^2}{\beta}$ | Diferenciales | Demografía de poblaciones. Dinámica de fluidos y partículas. |
| $Y = \alpha + \beta x$ | Lineales | Estadística. Respuesta a recursos limitantes |
| $Y = \alpha + \frac{\beta - \alpha}{1 + e^{\frac{\delta - x}{\lambda}}}$ | Logística | Crecimiento poblacional. Competencia. |
| $Y = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ | Gaussiana | Dispersión. Estadística. Contaminación. |
| $Y = \frac{\alpha}{\beta + x}$ | Hipérbola | Contaminación. |
| $Y = \alpha + \beta x + \delta x^2 + \gamma x^3$ | Polinomios | Dinámicas temporales. Estadística. Dinámica enzimática. |
| $Y = \alpha + \beta \times sen(x)$ | Trigonométricas | Dinámicas cíclicas |

**Tabla 3.** Pasos básicos para la escritura de una ecuación. Readaptado de Schneider 1994.

1. Establecer variable de respuesta;
2. Definir conceptualmente y asignar unidades a la variable de respuesta;
3. Establecer variable/s explicatorias;
4. Definir conceptualmente y asignar unidades a cada una;
5. Escribir la variable de respuesta como una función de las variables explicatorias:
   Respuesta = F (Expl1, Expl2, Expl3);
6. Escribir una ecuación razonando las cantidades e incluir una descripción empírica de la forma de la relación, como en los análisis exploratorios de datos;
7. Corroborar que las unidades en la ecuación se cancelen correctamente;
8. Realizar cálculos y contrastarlos contra datos y medidas independientes de ser posible;
9. Revisar la ecuación para incluir tantos procesos como sea necesario teniendo siempre presente el principio de parsimonia (simpleza).
10. Corroborar la coherencia de las unidades después de cada edición y antes de realizar los cálculos.

# Ecuaciones diferenciales

Aunque los matemáticos puros generalmente se interesan en la matemática por sí misma como un sistema o estructura, los ingenieros, los científicos de otras disciplinas y los matemáticos aplicados generalmente se interesan en el empleo de la matemática para explicar, describir, controlar o predecir el comportamiento de distintos tipos de fenómenos (e.g. físicos, químicos, biológicos, económicos). La más grande revolución en los fundamentos científicos y tecnológicos de nuestra civilización actual se debió a la aplicación del **cálculo** a la resolución de los problemas materiales. En efecto, gran parte del desarrollo del cálculo fue el resultado del esfuerzo de la humanidad para formular los problemas materiales en forma matemática de tal manera que los fenómenos relacionados pudieran describirse y comprenderse mejor. Galileo Galilei (1564-1642), famoso por sus aportes en la matemática, física, astronomía, filosofía y su célebre frase "eppur si muove" respecto de la Tierra y el conflicto con la Iglesia Católica, fue quien sentó las bases de la necesidad del cálculo. Posteriormente, a mediados del siglo XVII, Isaac Newton (1642-1727) y Göttfried Leibniz (1646-1716), quienes independientemente desarrollaron las ideas fundamentales del cálculo, fueron también los primeros en considerar las ecuaciones diferenciales. Por ejemplo, la bien conocida segunda ley de la mecánica de Newton puede expresarse por medio de una ecuación diferencial de segundo orden. Actualmente, gran parte de diferentes tipos de modelos analíticos relacionados con una diversidad de problemas, suelen representar a las variables de flujo a través de **ecuaciones diferenciales**.

## ¿Qué es una ecuación diferencial?

Una ecuación diferencial es aquella que contiene en su expresión una razón de cambio (i.e. derivadas / diferenciales) de una función desconocida. Las ecuaciones diferenciales (Ec. 1) pueden ser ordinarias o parciales dependiendo si dependen de una o más variables independientes y de distinto orden en función del grado de derivación.

$$\frac{dy}{dx} = f(y)$$

$$\frac{dy}{dx} = f'(x) = \frac{df(x)}{dx} = \lim_{\Delta x \to 0} \frac{f(x+\Delta x) - f(x)}{\Delta x} \qquad \textbf{[1]}$$

Por ejemplo, sabemos que *dy/dx* es la razón de cambio de *y* respecto de *x*. Supóngase que, por medio de experimentos y observaciones, podemos descubrir una expresión para esta razón de cambio. Podemos entonces establecer una ecuación que relacione a *x, y* y *dy/dx* y podemos estudiar esta ecuación para encontrar, si es posible, la relación directa entre *x* e *y*. Para ilustrarlo, supóngase que observamos la degradación de un agroquímico en un suelo. Podríamos notar que la velocidad a la que tiene lugar la degradación es proporcional a la cantidad presente del agroquímico en cada instante (podría concebirse esta idea, por ejemplo, mediante un gráfico de datos tomados durante un largo tiempo). Si *x* representa la cantidad del agroquímico presente en el tiempo *t*, entonces *dx*/*dt* representa la razón de cambio de *x* con respecto al tiempo. En consecuencia, la observación experimental puede formularse matemáticamente mediante la ecuación:

$$\frac{dx}{dt} = -kx$$

donde $k$ es una constante de proporcionalidad, y donde el signo negativo indica que la cantidad $x$ está decreciendo a medida que pasa el tiempo. Una ecuación de este tipo que implica una función desconocida (en este caso la concentración del agroquímico para un momento particular) y su derivada se llama **ecuación diferencial**. Una ecuación diferencial es frecuentemente una expresión de un problema físico en forma matemática. El problema de hallar la función desconocida, esto es, de resolver la ecuación diferencial, se convierte en un problema de técnica matemática.

## Algunas aplicaciones con ecuaciones diferenciales

Un modelo del crecimiento de una población se basa en la hipótesis de que *la velocidad de crecimiento de la población es proporcional al tamaño de ésta*. La razón de cambio del tamaño de la población, sólo depende de su tamaño y, en este modelo en particular, las limitaciones de espacio o recursos no tienen ningún efecto. Esta hipótesis es razonable para pequeñas poblaciones de organismos en grandes entornos como, por ejemplo, las primeras colonias de bacterias en algún medio o los primeros colonizadores de América del Norte.

Dada una hipótesis tan simple, esperamos que el modelo que la describe también lo sea. Las cantidades implicadas son:

- $t$ : tiempo (variable independiente)
- $P$ : tamaño de la población (variable dependiente), y
- $k$ : constante de proporcionalidad (parámetro) entre la tasa de crecimiento de la población y el tamaño de ésta.

El parámetro $k$ suele llamarse *coeficiente de velocidad de crecimiento.*

Las unidades para esas cantidades dependen de la aplicación. Si estamos modelando el crecimiento de moho en el pan, entonces $t$ podría medirse en *días* y $P(t)$ sería el *área cubierta por el moho* o bien el *peso del moho*. Si estamos hablando de la población europea en América, entonces $t$ probablemente se medirá en *años* y $P(t)$ en *millones de personas*. En este caso se haría corresponder $t = 0$ a cualquier tiempo que se desee.

Expresemos ahora la hipótesis usando esta notación. La tasa de crecimiento de la población es la derivada $dP/dt$. Puesto que ésta es proporcional al tamaño de la población, se expresa como el producto, $kP$, del tamaño de la población $P$ y la constante de proporcionalidad $k$. Por consiguiente, la hipótesis se expresa por la ecuación diferencial

$$\frac{dP}{dt} = kP$$

Este tipo de ecuación diferencial se denomina de **primer orden** porque contiene sólo primeras derivadas de la variable dependiente, y es una **ecuación diferencial ordinaria** porque no contiene derivadas parciales. Hemos escrito esta ecuación diferencial utilizando la notación de Leibniz, es decir, $dP/dt$, pero hay otras maneras de expresar la misma ecuación diferencial. En particular, también podríamos escribirla como $P' = kP$ o como $\dot{P} = kP$. Esta última notación (llamada notación "punto") suele utilizarse cuando la variable independiente es el tiempo.

Nos preguntamos qué predice el modelo planteado, es decir, qué nos dice acerca de la situación que se está modelando. Dado que $dP/dt = kP$ para alguna constante $k$, $dP/dt = 0$ si $P = 0$. Entonces, la función constante $P(t) = 0$ es una solución de la ecuación diferencial. A este tipo especial se le llama **solución de equilibrio** porque es constante para siempre. En términos del modelo de población, corresponde a una especie que ya no existe.

Si $P(t_0) \neq 0$ en algún tiempo $t_0$, entonces en el tiempo $t = t_0$

$$\frac{dP}{dt} = kP(t_0) \neq 0 .$$

En consecuencia, la población no es constante. Si $k > 0$ y $P(t_0) > 0$, obtenemos

$$\frac{dP}{dt} = kP(t_0) > 0$$

en el tiempo $t = t_0$ y la población está creciendo (como era de esperarse). Conforme $t$ crece, $P(t)$ aumenta, por lo que $dP/dt$ también aumenta haciendo que, a su vez $P(t)$ crezca aún más rápidamente. Es decir, la velocidad de crecimiento aumenta en relación directa con el tamaño de la población. Podemos esperar, por ende, que la gráfica de la función $P(t)$ tenga la forma que se muestra en la Figura 11.

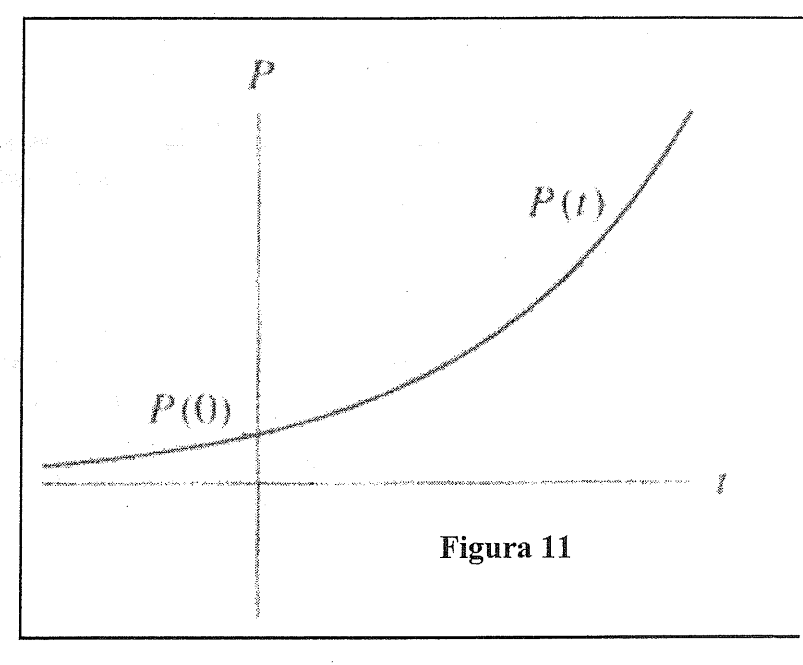


**Figura 11**

El valor de $P(t)$ en $t = 0$ se llama **condición inicial**. Si comenzamos con una condición inicial diferente obtenemos una función $P(t)$ distinta, como se indica en la Figura 12. Si $P(0)$ es negativa (recordando que $k > 0$), tenemos entonces que $dP/dt < 0$ para $t = 0$, por lo cual $P(t)$ inicialmente está disminuyendo. Al crecer $t$, $P(t)$ se vuelve más negativa. La imagen debajo del eje $t$ es una reflexión de la imagen que está por encima del mismo, aunque esto no tiene sentido biológico puesto que un tamaño de población no puede ser negativo.

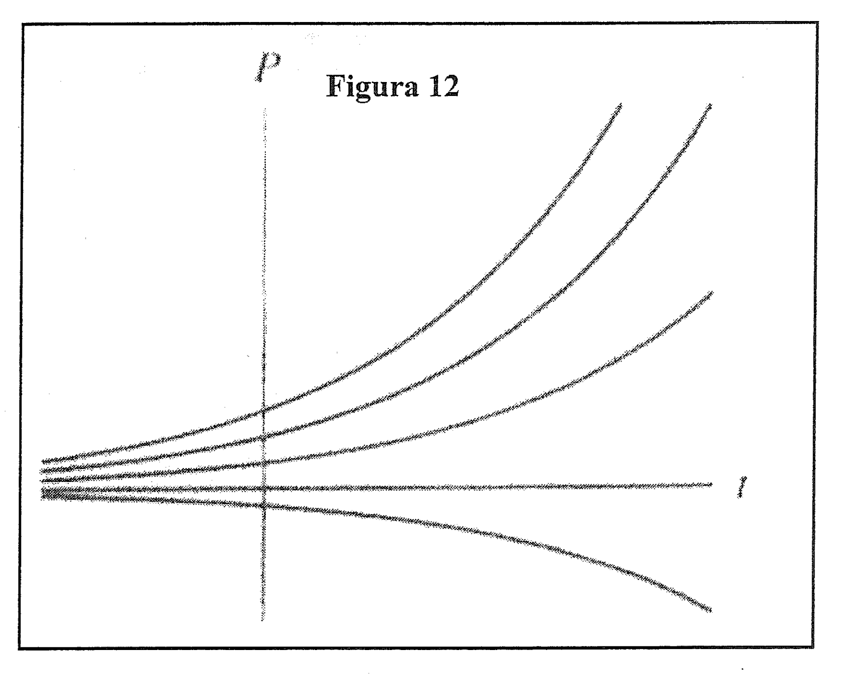


**Figura 12**

Nuestro análisis de la manera en que $P(t)$ crece cuando $t$ aumenta se denomina **análisis cualitativo** de la ecuación diferencial. Si todo lo que nos interesa es saber si el modelo predice "explosiones de población", entonces podríamos responder que "sí, siempre y cuando $P(0) > 0$".

## Soluciones analíticas

Si, por otra parte, conocemos el valor exacto, $P_0$, de $P(0)$ y queremos predecir el valor de $P(10)$ o $P(100)$, entonces necesitamos información más precisa sobre la función $P(t)$. El par de ecuaciones

$$\frac{dP}{dt} = kP \qquad \text{y} \qquad P(0) = P_0$$

se denomina **problema de valor inicial**. Y una **solución** al problema de valor inicial es una función $P(t)$ que satisface ambas ecuaciones. Es decir,

$$\frac{dP}{dt} = kP \text{ para toda } t \qquad \text{y} \qquad P(0) = P_0.$$

En consecuencia, para solucionar esta ecuación diferencial debemos hallar una función $P(t)$ cuya derivada sea el producto de $k$ con $P(t)$. Una manera (no muy sutil) de encontrarla es hacer una conjetura. En este caso, es relativamente fácil ver cuál es la forma correcta para $P(t)$ porque sabemos que la derivada de una función exponencial es, esencialmente, ella misma (podemos eliminar este proceso de conjeturar usando el método de separación de variables que veremos luego, pero por ahora ensayaremos el método exponencial y veamos a qué nos conduce). Después de un par de intentos con varias formas de dicha función, vemos que

$$P(t) = e^{kt}$$

su derivada, $dP/dt = ke^{kt}$, es el producto de $k$ con $P(t)$. Pero existen otras soluciones posibles, ya que $P(t) = Ce^{kt}$ (donde $C$ es una constante) da $dP/dt = C(ke^{kt}) = k(Ce^{kt}) = kP(t)$. Así, $dP/dt = kP$ para toda $t$ y para cualquier valor de la constante $C$.

Existe un número infinito de soluciones para la ecuación diferencial, uno para cada valor de $C$. Para determinar cuál de ésas es la correcta para la situación considerada, usamos la condición inicial dada. Tenemos

$$P_0 = P(0) = C \cdot e^{k \cdot 0} = C \cdot e^0 = C \cdot 1 = C.$$

En consecuencia debemos escoger $C = P_0$, por lo que una solución del problema del valor inicial es

$$P(t) = P_0 \cdot e^{kt}.$$

Hemos obtenido una fórmula para nuestra solución, no solamente una imagen cualitativa de su gráfica.

La función $P(t)$ se llama **solución al problema de valor inicial** así como **solución particular** de la ecuación diferencial. El conjunto de funciones $P(t) = C \cdot e^{kt}$ se llama **solución general** de la ecuación diferencial, porque podemos usarla para encontrar la respuesta particular correspondiente a cualquier problema de valor inicial.

**Ejemplo 1. La población de los Estados Unidos.** Para ejemplificar cómo puede usarse este modelo, consideremos las cifras de los censos de los Estados Unidos desde 1790 que se presentan en la Tabla 4. Veamos qué tan bien se ajusta el modelo de crecimiento ilimitado a estos datos. Medimos el tiempo en años y la población $P(t)$ en millones de personas. Hacemos que $t = 0$ sea el año 1790, por lo que la condición inicial es $P(0) = 3.9$. El problema correspondiente al valor inicial

$$\frac{dP}{dt} = kP, \qquad P(0) = 3.9,$$

tiene $P(t) = 3.9 \cdot e^{kt}$ como solución. Pero no podemos usar este modelo para hacer predicciones porque no conocemos el valor de $k$. Sin embargo, sabemos que la población en el año 1800 era de 5.3 millones y podemos usar este valor para determinar $k$. Si hacemos

$$5.3 = P(10) = 3.9 \cdot e^{k \cdot 10}$$

obtenemos

$$e^{k \cdot 10} = \frac{5.3}{3.9} \Rightarrow 10 \cdot k = \ln\left(\frac{5.3}{3.9}\right) \Rightarrow k \approx 0.03067$$

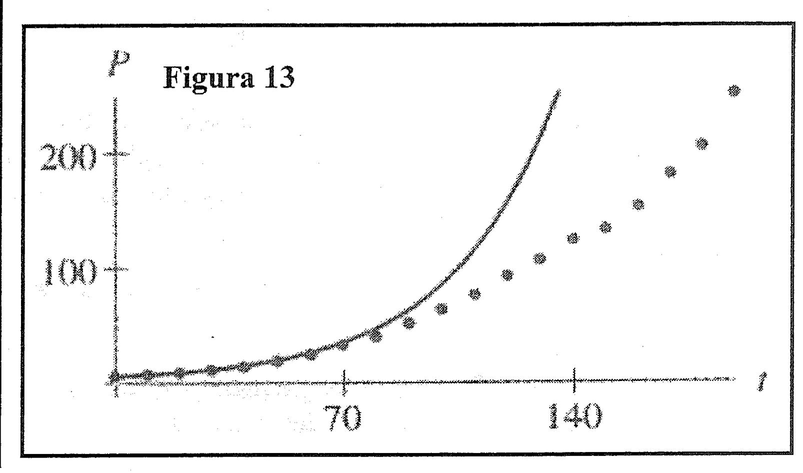
**Tabla 4.** Cifras de los censos de los Estados Unidos, en millones de personas.

| Año | $T$ | Real | $P(t) = 3.9 \cdot \exp(0.03067 \cdot t)$ |
|---|---|---|---|
| 1790 | 0 | 3.9 | 3.9 |
| 1800 | 10 | 5.3 | 5.3 |
| 1810 | 20 | 7.2 | 7.2 |
| 1820 | 30 | 9.6 | 9.8 |
| 1830 | 40 | 12 | 13 |
| 1840 | 50 | 17 | 18 |
| 1850 | 60 | 23 | 25 |
| 1860 | 70 | 31 | 33 |
| 1870 | 80 | 38 | 45 |
| 1880 | 90 | 50 | 62 |
| 1890 | 100 | 62 | 84 |
| 1900 | 110 | 75 | 114 |
| 1910 | 120 | 91 | 155 |
| 1920 | 130 | 105 | 210 |
| 1930 | 140 | 122 | 286 |
| 1940 | 150 | 131 | 388 |
| 1950 | 160 | 151 | 528 |
| 1960 | 170 | 179 | 717 |
| 1970 | 180 | 203 | 975 |
| 1980 | 190 | 226 | 1320 |
| 1990 | 200 | 249 | 1800 |
| 2000 | 210 | | 2450 |
| 2010 | 220 | | 3320 |
| 2020 | 230 | | 4520 |
| 2030 | 240 | | 6140 |
| 2040 | 250 | | 8340 |
| 2050 | 260 | | 11300 |

Nuestro modelo predice entonces que la población de los Estados Unidos está dada por

$$P(t) = 3.9 \cdot \exp(0.03067 \cdot t).$$

Como podemos ver en la Figura 13, este modelo de $P(t)$ predice razonablemente bien el tamaño de la población hasta 1860, pero después de este año la predicción resulta muy grande (la **Tabla 4** incluye una comparación de los valores predichos con los datos reales).



Nuestro modelo es bastante bueno siempre que la población sea relativamente pequeña pero, con el paso del tiempo, el modelo predice que la población continuará creciendo sin límite y, obviamente, esto no sucede en el mundo real. En consecuencia, si queremos un modelo que sea exacto sobre una escala grande de tiempo, debemos tomar en cuenta el hecho de que las poblaciones existen en una cantidad finita de espacio y con recursos limitados.

## Modelo logístico de la población

Para ajustar el modelo de crecimiento exponencial de la población que tome en cuenta un entorno y recursos limitados, agregamos las siguientes hipótesis:

- Si la población es pequeña, la razón de crecimiento de la población es proporcional a su tamaño.
- Si la población es demasiado grande como para ser soportada por su entorno y los recursos, entonces su tamaño disminuirá, es decir, la razón de crecimiento será negativa.

Para este modelo usamos otra vez

- $t$ : tiempo (variable independiente)
- $P$ : tamaño de la población (variable dependiente), y
- $k$ : coeficiente de la razón de crecimiento para poblaciones pequeñas (parámetro).

Pero la hipótesis acerca de recursos limitados introduce otra cantidad: el tamaño de la población que ha de considerarse como "demasiado grande". Esta cantidad es un segundo parámetro, denotado $N$, que llamaremos **capacidad de carga** del entorno. En términos de la capacidad de carga, estamos suponiendo que $P(t)$ crece si $P(t) < N$, pero si $P(t) > N$ supondremos que está decreciendo.

Usando esta notación, podemos rescribir nuestras hipótesis como:

- $\frac{dP}{dt} \approx kP$ si $P$ es pequeña (primera hipótesis)
- Si $P > N$, $\frac{dP}{dt} < 0$ (segunda hipótesis)

También queremos que el modelo sea "algebraicamente simple" o, por lo menos, tan simple como sea posible, por lo cual tratamos de modificar el modelo exponencial lo menos posible. Por ejemplo, podríamos intentar una expresión de la forma

$$\frac{dP}{dt} = k \cdot \Psi \cdot P$$

donde $\Psi$ sea cercano a 1 si $P$ es pequeña y que sea negativo si $P > N$. La expresión más simple que tiene estas propiedades es la función

$$\Psi = 1 - \frac{P}{N}.$$

Nótese que esta expresión es igual a 1 si $P = 0$ y es negativa si $P > N$. Nuestro modelo es entonces

$$\frac{dP}{dt} = k \cdot \left(1 - \frac{P}{N}\right) \cdot P.$$

Este modelo se llama **modelo logístico de la población** con velocidad de crecimiento $k$ y capacidad de carga $N$. Se trata de otra ecuación diferencial de primer orden. Se dice que esta ecuación es **no lineal** porque su lado derecho no es una función lineal de $P$ como lo era en el modelo de crecimiento exponencial.

### *Análisis cualitativo del modelo logístico*

Aunque la ecuación diferencial logística es ligeramente más complicada que la del modelo decrecimiento exponencial, no hay modo de que podamos conjeturar soluciones. El método de separación de variables que analizaremos después, produce una fórmula para la solución de esta ecuación diferencial particular. Pero por ahora nos apoyaremos meramente en método cualitativos para ver qué anticipa este modelo a largo plazo.

Primero sea

$$f(P) = k \cdot \left(1 - \frac{P}{N}\right) \cdot P$$

el lado derecho de la ecuación diferencial. En otras palabras, la ecuación diferencial puede escribirse como

$$\frac{dP}{dt} = f(P) = k \cdot \left(1 - \frac{P}{N}\right) \cdot P$$

Podemos obtener información cualitativa sobre las soluciones a la ecuación diferencial si sabemos cuándo $dP/dt$ es 0, dónde es positiva y dónde es negativa.

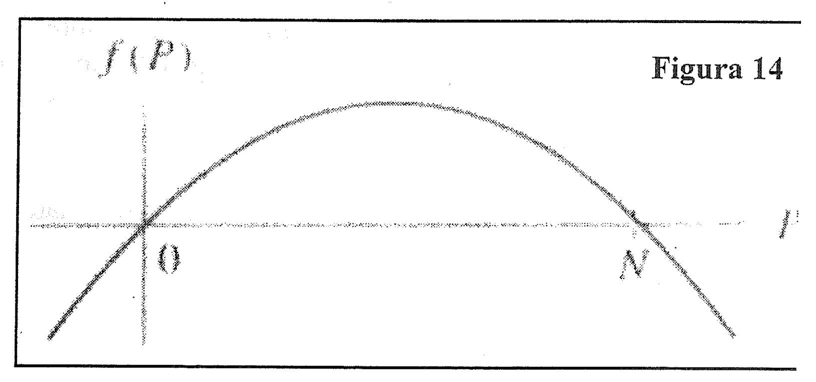
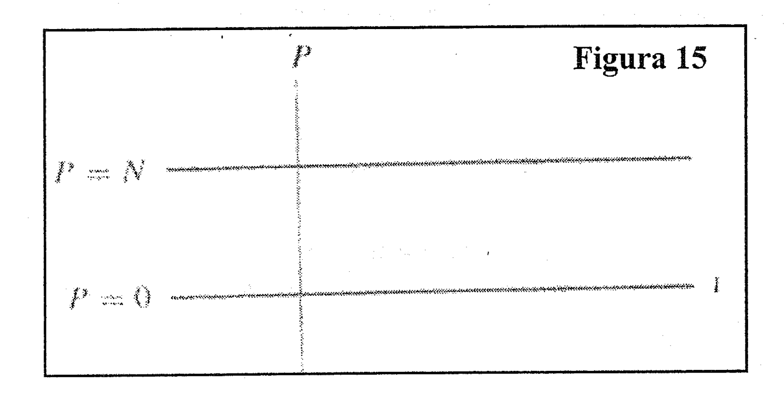
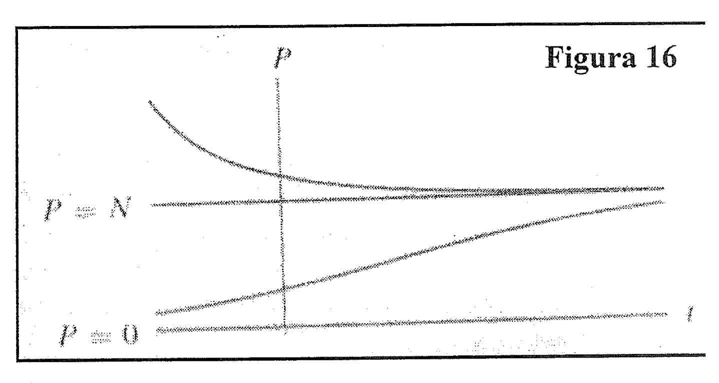


**Figura 14**

Si trazamos la gráfica de la función cuadrática $f$ (Figura 14), observamos que ella corta al eje $P$ en exactamente dos puntos, $P = 0$ y $P = N$. En esos casos, $dP/dt = 0$. Como la derivada de $P$ desaparece para toda $t$, la población permanece constante para si $P = 0$ o $P = N$. Es decir, las funciones constantes $P(t) = 0$ y $P(t) = N$ resuelven la ecuación diferencial. Esas dos soluciones constantes tienen mucho sentido: si el tamaño de la población es igual a 0, permanecerá en 0 indefinidamente (inmigraciones excluidas); si el tamaño de la población es exactamente igual al asociado con la capacidad de carga, no crecerá ni disminuirá. Igual que antes, decimos que $P = 0$ y $P = N$ son **puntos de equilibrio**. Las funciones constantes $P(t) = 0$ y $P(t) = N$ se denominan **soluciones de equilibrio** (Figura 15).

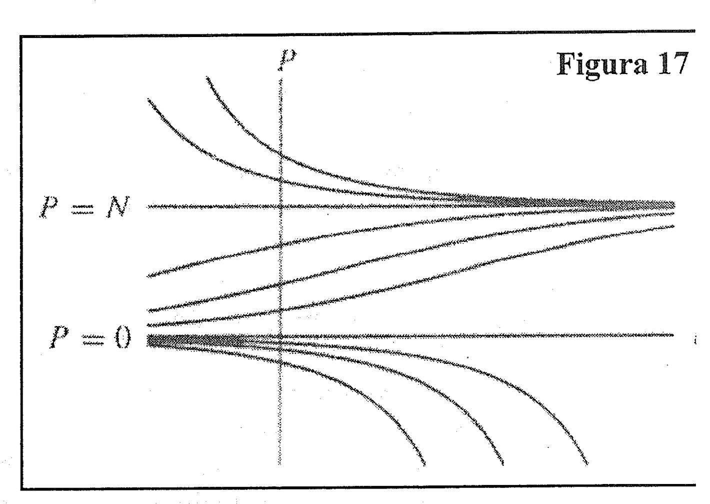



El comportamiento a largo plazo de la población es muy diferente para otros valores. Si la población inicial se encuentra entre 0 y $N$, tenemos entonces que $f(P) > 0$. En este caso, la tasa de crecimiento $dP/dt = f(P)$, es positiva y, en consecuencia, el tamaño de la población $P(t)$ está creciendo. En tanto que $P(t)$ se encuentre entre 0 y $N$, la población continúa incrementándose, pero cuando va llegando a la capacidad de carga, $N$, $dP/dt = f(P)$ se acerca a 0, por lo que se espera que el tamaño de la población se estabilice cuando tienda al valor $N$ ( Figura 16).




Si $P(0) > N$, entonces, $dP/dt = f(P) < 0$ y la población estará disminuyendo. Y cuando tiende a la capacidad de carga $N$, $dP/dt$ se aproxima a 0 y esperamos, otra vez, que el tamaño de la población se estabilice en $N$. Finalmente – y esto es sólo de interés matemático puesto que no tiene sentido biológico – si $P(0) < 0$, tenemos también que $dP/dt = f(P) < 0$. Vemos otra vez que $P(t)$ disminuye, pero ahora no se estabiliza en ningún valor particular ya que $dP/dt$ se vuelve más y más negativa conforme $P(t)$ decrece.

Así, solamente partiendo de la gráfica de $f$, podemos esbozar varias diferentes soluciones con condiciones iniciales diferentes, todas sobre los mismos ejes. La única información que necesitamos es el hecho de que $P = 0$ y $P = N$ son soluciones de equilibrio; $P(t)$ crece si $0 < P < N$, y $P(t)$ disminuye si $P > N$ o $P < 0$. Por supuesto, los valores exactos de $P(t)$ en cualquier tiempo dado $t$, dependerán de los valores de $P(0)$, de $k$ y de $N$ (Figura 17).



## Sistemas presa - depredador

Ninguna especie vive aislada y las interacciones entre especies proporcionan algunos de los modelos más interesantes por estudiar. Presentamos ahora un sistema simple presa-predador de ecuaciones diferenciales donde los organismos de una especie se alimentan de los de la otra. La diferencia más obvia entre éste y los modelos previos es que tenemos **dos** cantidades que dependen del tiempo. Supongamos que los organismos presa son conejos ($C$) y los depredadores, zorros ($Z$). $Z$ y $C$ son los tamaños de las poblaciones de zorros y conejos, respectivamente. Las hipótesis de nuestro modelo son:

- si no hay zorros presentes, los conejos se reproducen a una tasa proporcional al tamaño de su población y no les afecta la superpoblación.
- los zorros se comen a los conejos y la tasa a la que los conejos son devorados es proporcional a la tasa a la que los zorros y los conejos interactúan.
- sin conejos que comer, la población de zorros declina a una tasa proporcional a su tamaño.
- la tasa de nacimientos de los zorros va en proporción al número de conejos comidos por zorros que, por la segunda hipótesis, es proporcional a la tasa a la que los zorros y conejos interactúan.

Para formular este modelo en términos matemáticos, necesitamos cuatro parámetros adicionales a nuestra variable independiente $t$ y a nuestras dos variables dependientes $Z$ y $C$. Los parámetros son:

$\alpha$ : coeficiente correspondiente a la tasa de crecimiento de la población de conejos;
$\beta$ : constante de proporcionalidad que mide el número de interacciones conejos-zorros en las que el conejo es devorado;
$\gamma$: coeficiente correspondiente a la tasa de muertes de zorros;
$\delta$ : constante de proporcionalidad que mide el beneficio a la población de zorros de un conejo devorado.

Cuando formulamos nuestro modelo, seguimos la convención de que $\alpha$, $\beta$, $\gamma$ y $\delta$ son todos positivos.

Nuestras hipótesis primera y tercera son similares a la que plantea el modelo de crecimiento ilimitado que ya hemos visto. En consecuencia, ellos dan términos de la forma $\alpha C$ en la ecuación $dC/dt$ y $-\gamma Z$ (puesto que la población de zorros declina) en la ecuación para $dZ/dt$.

La tasa a la que los conejos son devorados es proporcional a la tasa de interacción entre zorros y conejos, por lo cual necesitamos un término que modele la razón de interacción de ambas poblaciones: que crezca si $C$ o $Z$ aumentan, pero que desaparezca si $C = 0$ o $Z = 0$. Una notación que incorpora esas hipótesis es $CZ$. Así, modelamos los efectos de las interacciones conejo-zorro sobre $dC/dt$ por medio de un enunciado de la forma $-\beta CZ$. La cuarta hipótesis da un término similar en la ecuación para $dZ/dt$. En este caso, cazar conejos ayuda a los zorros, por lo que añadimos un término de la forma $\delta CZ$.

Al plantear estas hipótesis, obtenemos el modelo $\begin{cases} \dfrac{dC}{dt} = \alpha C - \beta CZ \\ \dfrac{dZ}{dt} = -\gamma Z + \delta CZ \end{cases}$.

Consideradas juntas, este par de expresiones conforman un **sistema de ecuaciones diferenciales ordinarias de primer orden** (contiene más de una variable independiente, pero sólo derivadas primeras). Se dice que el sistema es **acoplado** porque las tasas de cambio de $C$ y $Z$ dependen tanto de $C$ como de $Z$.

Es importante notar los signos de los términos en este sistema. Dado que $\beta > 0$, el término -$\beta CZ$ es no positivo, por lo que un incremento en el número de zorros disminuye la razón de crecimiento de la población de conejos. Además, dado que $\delta > 0$, el término $\delta CZ$ es no negativo. En consecuencia, un incremento en el número de conejos incrementa la tasa de crecimiento de la población de zorros.

Aunque este modelo puede parecer relativamente simple, ha sido la base de muchos interesantes estudios ecológicos. En particular, Volterra y D'Ancona usaron el modelo con éxito para explicar el incremento de la población de tiburones en el mar Mediterráneo durante la Primera Guerra Mundial, cuando la pesca de las especies presa de los tiburones, decreció. El modelo puede también usarse como base para el estudio de los efectos de los pesticidas en la población de insectos depredadores e insectos presa.

Una **solución** para este sistema de ecuaciones es, a diferencia de los modelos previos, un par de funciones $C(t)$ y $Z(t)$, que describen las poblaciones de conejos y zorros como funciones del tiempo. Como el sistema es acoplado, no podemos determinar cada una de esas funciones en forma aislada. Más bien, debemos resolver ambas ecuaciones diferenciales en forma simultánea. Desafortunadamente, para la mayor parte de los valores de los parámetros, es imposible determinar explícitamente fórmulas para $C(t)$ y $Z(t)$. Esas funciones no pueden expresarse en términos de funciones conocidas tales como polinomios, senos, cosenos, exponenciales y otras parecidas. Pero, como veremos más adelante, esas soluciones existen, aunque no hay esperanza de encontrarlas jamás con exactitud. Visto los métodos analíticos para resolver este sistema están destinados a fallar, se deben usar procedimientos cualitativos o numéricos para "hallar" $C(t)$ y $Z(t)$.

## Los enfoques analítico, cualitativo y numérico

Nuestro análisis de los modelos de esta sección ilustra tres enfoques diferentes para el estudio de las soluciones de las ecuaciones diferenciales. El enfoque **analítico** busca fórmulas explícitas que describan el comportamiento de las soluciones a las cuales pueden llegarse mediante operaciones algebraicas o integrando directamente. Vimos aquí que las funciones exponenciales dan soluciones explícitas al modelo de crecimiento exponencial. Desafortunadamente, un gran número de ecuaciones importantes no pueden tratarse con el método analítico; simplemente no hay manera de encontrar la expresión exacta que describa la situación porque la integral correspondiente no se puede resolver. Nos vemos entonces forzados a recurrir a métodos alternativos.

Un procedimiento particularmente poderoso para describir el comportamiento de las soluciones es el enfoque **cualitativo** que se basa en el empleo de la geometría para obtener un panorama del comportamiento del modelo, tal como lo hicimos con el modelo logístico de crecimiento de la población. No se utiliza para dar valores precisos de la solución en tiempos específicos sino para determinar su comportamiento a largo plazo. Con frecuencia, ésta es justamente la clase de información que se requiere.

El tercer enfoque es el **numérico**. La computadora aproxima la solución que se busca. Aunque todavía no se ilustró ninguna técnica de aproximación numérica, más adelante veremos que son muy poderosas para obtener ideas respecto de las soluciones que se requieren.

Los tres métodos tienen sus ventajas y desventajas y una de las principales tareas al estudiar las ecuaciones diferenciales será determinar qué método, o combinación de métodos, funciona bien en cada caso específico.

# Evaluación de los Modelos: Calibración, Análisis de Sensibilidad y Validación

La última etapa en el desarrollo de un modelo requiere la corroboración de los resultados del mismo y una evaluación exhaustiva de su funcionamiento. Esta etapa de evaluación permitirá realizar la "puesta a punto" del mismo y conocer en detalle sus fortalezas y debilidades como herramienta predictiva. Esta última etapa podemos dividirla en tres partes diferentes: calibración, análisis de sensibilidad y validación.

La *calibración* permite ajustar los parámetros que determinan el comportamiento de un modelo de modo de obtener resultados probables de acuerdo con el conocimiento de ese sistema. La calibración permite para un modelo nuevo restringir el comportamiento del mismo sólo a las situaciones esperables desde lo que se conoce del sistema. Muchas veces varios procesos de calibración son necesarios cuando un modelo se emplea en diversas situaciones. Estos cambios pueden generar comportamientos inesperados del modelo que podrían corregirse a partir de nuevas parametrizaciones. Una manera muy utilizada de calibrar un modelo es correr el mismo para todo el rango posible de cada uno de sus parámetros. En algunos casos la cantidad de parámetros hace que todas las combinaciones sean imposibles de explorar en tiempos razonables y en estos casos existen técnicas que permiten optimizar la exploración del hiper-espacio paramétrico. Una vez obtenido los resultados de todas las simulaciones, estos son almacenados en bases de datos junto con sus parámetros y a través de consultas se pueden filtrar sólo aquellas situaciones que tienen correlato con el sistema modelado (Tabla 5).

**Tabla 5.** Tabla resumen con un registro por simulación con los valores de los parámetros correspondientes a cada corrida y sus respectivos resultados. La última columna establece los criterios de restricción según lo observado en el sistema real.

| Simulación | ***Parámetros*** | | | | | | ***Resultados*** | |
|---|---|---|---|---|---|---|---|---|
| | ***P1*** | P2 | P3 | P4 | P5 | P6 | Simulados | Filtro s/ Obs. |
| 1 | 2.3 | 3.5 | 4.5 | 122 | 25 | 10 | **52** | **30< y <36** |
| 2 | 2.6 | 3.5 | 4.5 | 122 | 25 | 10 | **48** | **30< y <36** |
| 3 | 2.9 | 3.5 | 4.5 | 122 | 25 | 10 | **33** | **30< y <36** |
| . | . | . | . | . | . | . | . | . |
| . | . | . | . | . | . | . | . | . |
| . | . | . | . | . | . | . | . | . |
| . | . | . | . | . | . | . | . | . |
| ***n*** | . | . | . | . | . | . | . | . |

El *análisis de sensibilidad* permite explorar las variaciones en los resultados del modelo frente a las variaciones en sus distintos parámetros. Este análisis permite conocer frente a qué parámetros el modelo es más sensible y por lo tanto cambios de pequeña magnitud generan grandes diferencias en sus salidas y cuales parámetros son más estables y por lo tanto los resultados no cambian a pesar de sus variaciones. Esta técnica permite entonces identificar cuales parámetros regulan en mayor medida el comportamiento del modelo y cuales podrían descartarse o ser replanteados. De este modo, el análisis de sensibilidad permite continuar con el desarrollo del modelo mejorando la implementación de

aquellos parámetros claves y descartando aquellos que no aportan o son redundantes. La implementación del análisis de sensibilidad requiere igual que en el caso anterior realizar un número grande de simulaciones que permitan explorar el rango de variación posible de cada parámetro. Una vez obtenidos los resultados de estas simulaciones a través de análisis de regresión entre una variable de respuesta predicha por el modelo y el parámetro se puede establecer su sensibilidad observando la prueba de hipótesis y la estimación de los coeficientes $\beta_1$ del análisis de regresión (Figura 18).

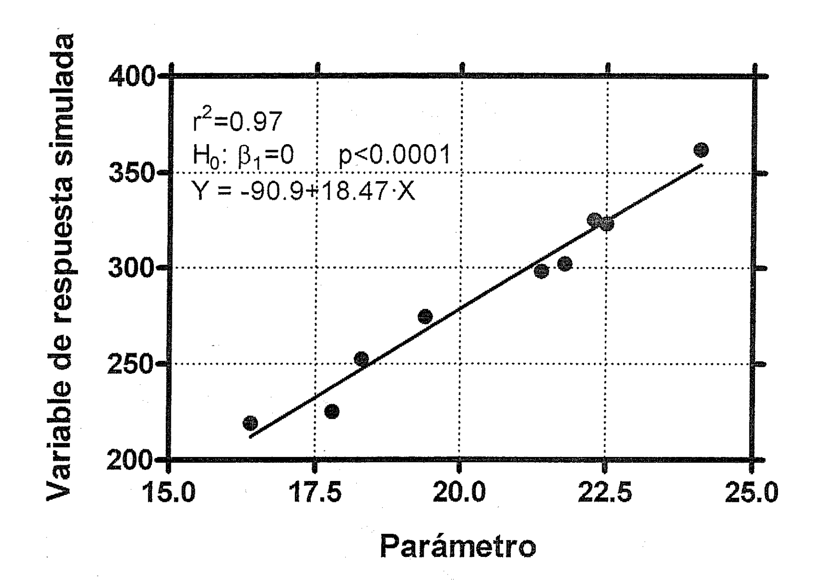



**Figura 18.** Relación entre un parámetro particular y la variable de respuesta simulada por un modelo. Se indica el coeficiente de determinación, la prueba de hipótesis para $\exists_1$ y la recta de mejor ajuste de acuerdo con el análisis de regresión lineal realizado.

Finalmente, la *validación* permite contrastar los datos reales con los simulados por el modelo. En este punto es importante distinguir entre la validación de la capacidad de extrapolación en el uso de un modelo que obedece a razones teóricas basadas en el conocimiento de los sistemas y a la validación del comportamiento del modelo para la región y bajo los supuestos que fue desarrollado que es estrictamente empírica. Este último proceso permite entonces conocer el grado de éxito en las predicciones efectuadas con el modelo de acuerdo con su proximidad a la realidad conocida. El modo de validar un modelo consiste en conseguir información real de las variables de ingreso que condicionan el funcionamiento del modelo y sus respectivas variables de respuesta para el sistema que fue modelado. Es importante que se trate de información distinta a la que se empleó para generar las reglas del modelo, de modo de no realizar construcciones circulares que invalidan el análisis. Una vez obtenido este conjunto de información, se pueden realizar simulaciones con el modelo utilizando el conjunto de parámetros calibrados y las mismas variables de ingreso con los mismos valores para el sistema real y contrastar los resultados (Figura 19).

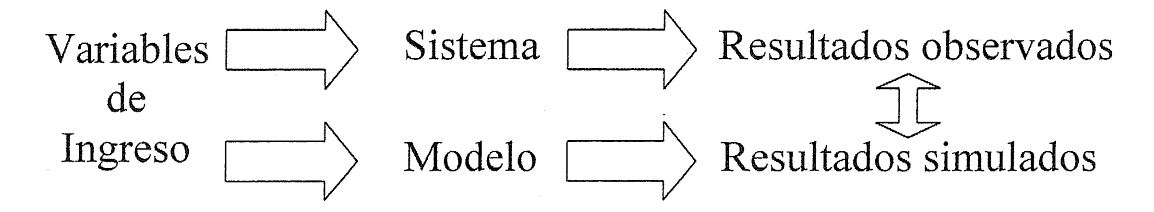



**Figura 19.** Esquema gráfico del proceso de validación.

En este punto es importante aclarar que aquellos modelos denominados "realistas" son más fáciles de validar, dado que la similitud conceptual entre los parámetros del modelo y sus resultados con las situaciones reales es muy cercana. Con esta información se puede graficar la relación entre los valores observados y simulados (Figura 20) y además calcular una medida de error muy simple que relaciona a ambos,

$$E = \sqrt{\frac{\sum (Sim - Obs)^2}{n}}$$

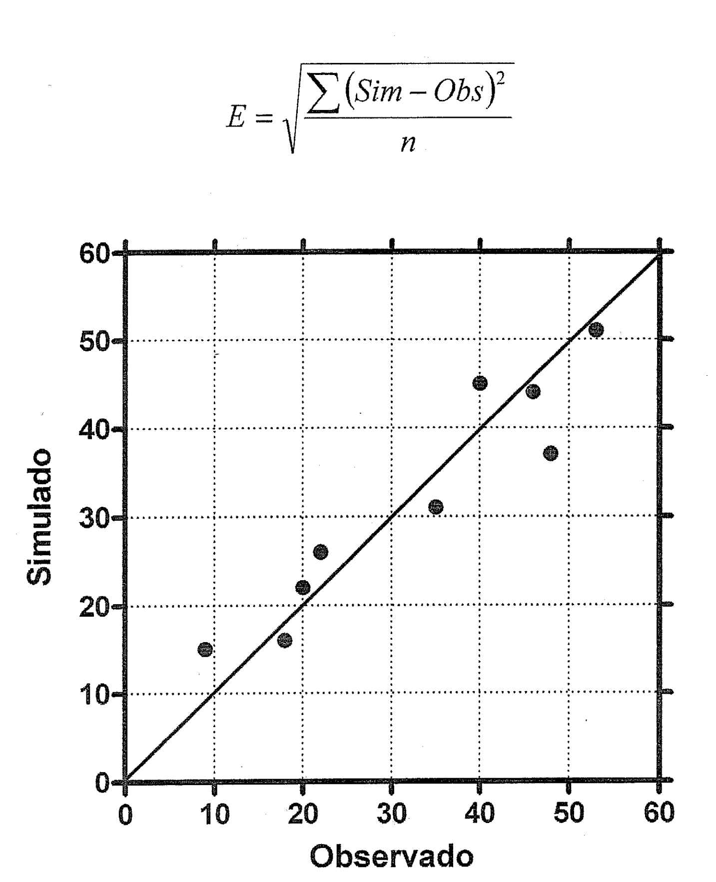


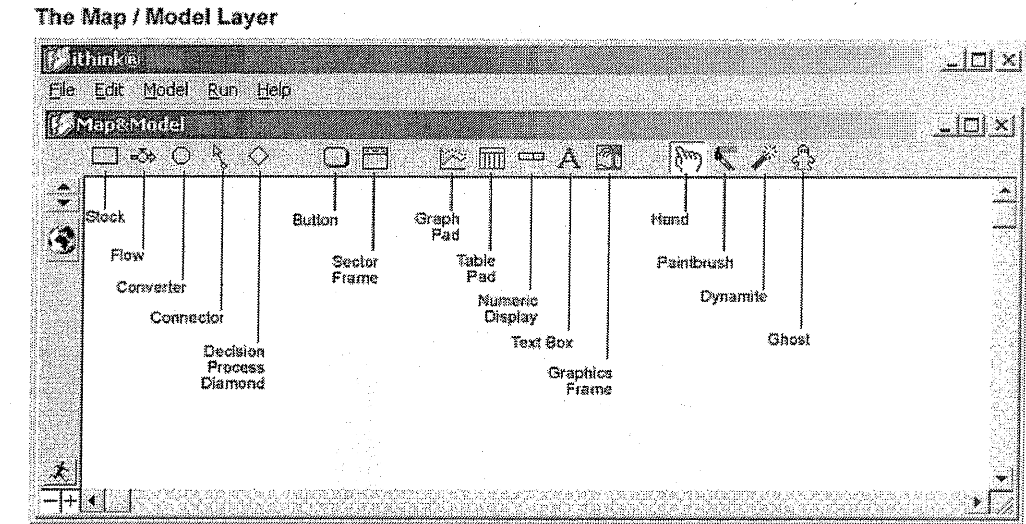

**Figura 20.** Relación entre los valores observados en un sistema real y los simulados por un modelo para las mismas condiciones. La recta indica la línea 1:1.

# Modelando en Stella©

Cómo se vio anteriormente, los lenguajes de programación (e.g. Delphi, C++, SmallTalk) son una herramienta apropiada para la construcción de modelos de simulación. Sin embargo, en la mayoría de los casos su uso queda restringido a programadores avezados con un conocimiento profundo de su funcionamiento y sintaxis, que generalmente los profesionales de las áreas biológicas no poseen. Para saldar esta situación, durante los últimos tiempos han aparecido ambientes de simulación que facilitan el modelado para un público con menos conocimientos de programación. Entre otros, ambientes como Stella, Vensim y Simulink, facilitan la programación de modelos basados en objetos simples y comunes a muchos problemas (e.g. variables de estado y flujo) y sin necesidad de entrar en cuestiones vinculadas a la programación tradicional como definir una variable, asignar memoria, programar bucles recursivos, etc. De este modo, el modelador puede concentrarse en el problema de interés y dejar para otras experiencias la cuestión informática. Es importante recalcar que si bien ambientes como Stella© pueden ser muy útiles y facilitar el modelado, el hecho de trabajar con programas "enlatados"[4] puede limitar las capacidades de lo que se puede hacer, en definitiva restringen la versatilidad para desarrollar un modelo a un determinado problema en particular. A continuación se brinda una guía rápida para modelar en Stella©.

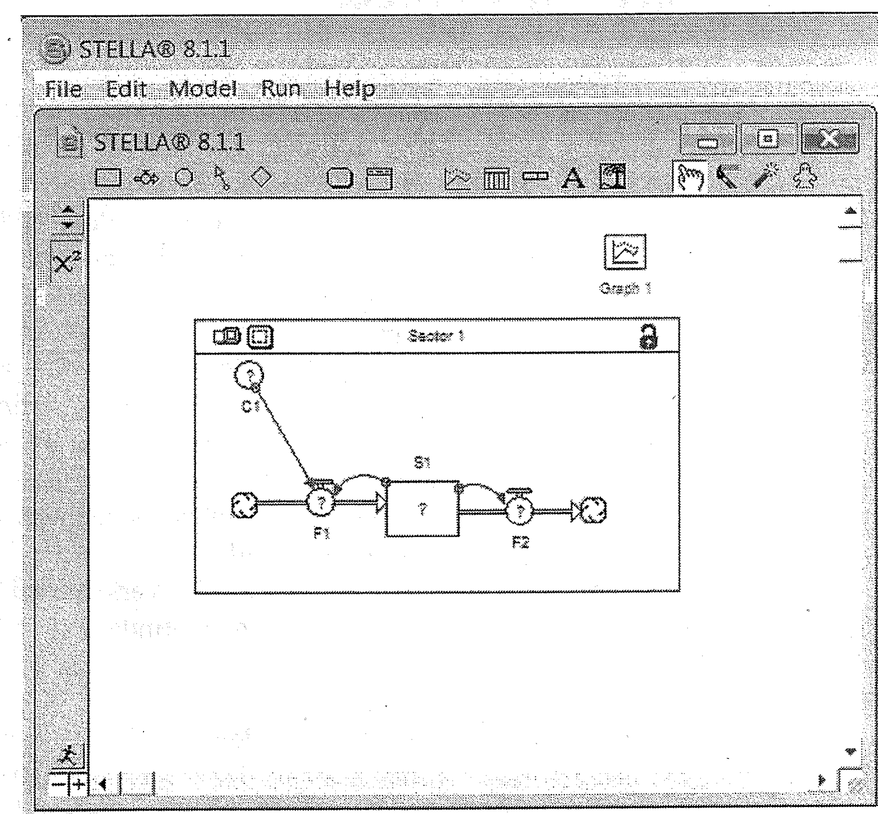
Stella© permite construir modelos analíticos basados en ecuaciones diferenciales, si bien se pueden hacer muchas cosas bajo este ambiente, también representa un límite para otros tipos de modelos. Este ambiente está organizado en torno a tres áreas de trabajo o estratos, nombradas como "Interface", "Map&Model" y "Equations". Podemos decir que la más importante es el área de trabajo donde se define el modelo, llamada como "Map&Model". En esta área existe una primera pantalla con una barra de menús donde se definen los objetos a utilizar para esto. Allí aparecen iconos que permiten agregar al área del mapa del modelo: variables de estado y variables de flujo (de distinto tipos), convertidores, conectores y procesos de decisión. El resto de los iconos permiten agregar tablas o figuras con resultados, botones de ejecución, sub-sectores dentro del modelo, fotos ilustrativas, cuadros de texto, etc. De este modo, y a través de pictogramas como cajas o válvulas se agregan en el área del mapa del modelo los mismos y se construye así el modelo de simulación de un modo gráfico / pictórico. También en esta etapa permite agregar breves párrafos de texto que explican cada una de las variables.

**The Map / Model Layer**

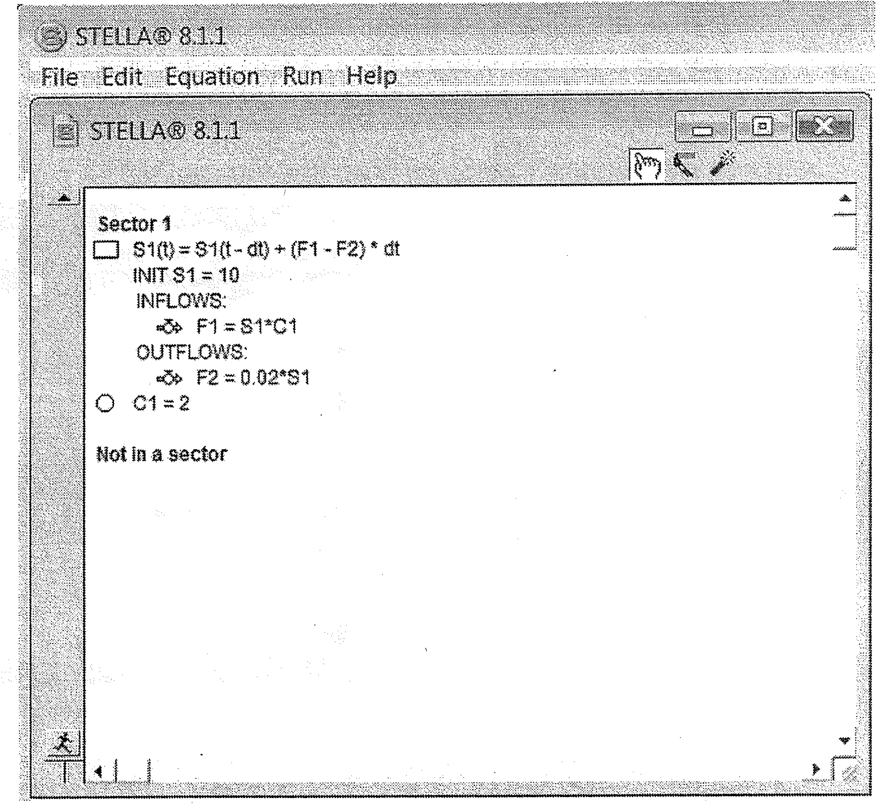



[4] Se refiere a programas sin posibilidad de modificación por el usuario.

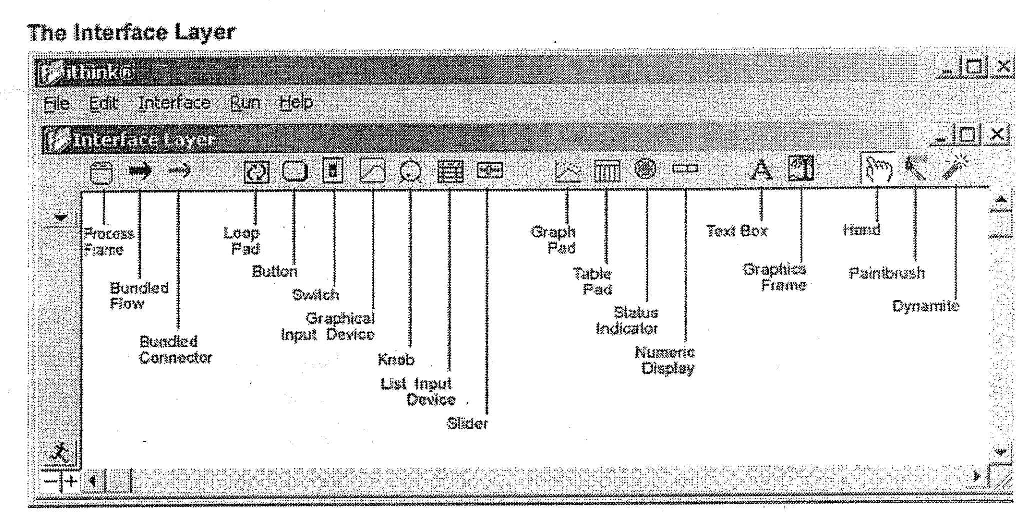
Una vez que el primer esquema o mapa del modelo está armado, dentro de la misma área se pasa a las definiciones formales de las reglas, funciones matemáticas, lógicas o estadísticas que caracterizan a las variables de flujo y convertidores previamente definidos en el mapa y se asignan los valores iniciales de las variables de estado. Aquí Stella posee una amplia librería con funciones matemáticas, estadísticas y lógicas estándares que facilitan la construcción de las reglas de funcionamiento del modelo. Además, permite leer datos desde planillas de cálculo, definir las unidades de las variables, la dinámica de resolución del modelo al definir el *time-step* o mínima resolución temporal y el método analítico de resolución entre otras facilidades.




Otra área de importancia en Stella© es "Equations", donde el ambiente por sí mismo genera las ecuaciones formales que se definieron gráficamente en el área de "Map&Model". Allí se verá al principio las ecuaciones diferenciales que rigen el comportamiento de las variables de estado del modelo y luego las funciones ingresadas a partir de las librerías para las variables de flujo o convertidores. Uno podría programar el modelo desde está área si lo deseará. Una vez que el modelo está definido completamente, se pueden definir otros aspectos de la simulación y correr el modelo para evaluar su correcto funcionamiento. Incluso para esto, Stella también brinda una serie de opciones para realizar análisis de sensibilidad, de modo de ejecutar el modelo para varias veces para distintas parametrizaciones.



Finalmente y una vez que el modelo está terminado y su correcto funcionamiento comprobado, la última área o estrato de interés "Interface" tiene que ver con agilizar y facilitar la interfase del usuario. Esta es un área que generalmente se deja para el final y tiene que ver con el embellecimiento del modelo para hacerlo fácilmente disponible a cualquier usuario sin necesidad de conocimientos previos y profundos en el mismo. Para esto se pueden agregar cuadros de texto, dibujos, fotos o esquemas con explicaciones del modelo y además objetos como botones, pulsadores, barras de desplazamiento entre otros que faciliten los cambios en los parámetros.




Botón de ejecución que da inicio a la simulación

Figura con resultados de la simulación

Perillas de variación de los parámetros del modelo

# Bibliografía modelos de simulación


Austin, A. T. 2002. Differential effects of precipitation on production and decomposition along a rainfall gradient in Hawaii. *Ecology* 83:328-338.

Austin, A. T., y Vitousek P. M. 1998. Nutrient dynamics on a precipitation gradient in Hawaii. *Oecologia* 113:519-529.

Clark J.S., Carpenter S.R., BarberM., Collins S., Dobson A., Foley J., Lodge D., Pascual M., Pielke R.,Pizer W. Jr., Pringle C., Reid W., Rose K., Sala O., Schlesinger W., Wall D. and Wear D. (2001). Ecological forecasting: an emerging imperative. *Science* 293: 657-660.

Dunham, A.E. (1993). Population responses to environmental change: Operative environments, physiologically strucured models, and population dynamics. pp: 95-119. En Biotic Interactions and Global Change., Eds. Kareiva, P. M., J. G. Kingsolver, R.B. Huey, (1993) Sinauer Associates Inc., pp. 559.

Dunning J.B., Stewart D.J., Danielson B.J., Noon B.R., Root T.L., Lambertson R.H. and Stevens E.E. (1995). Spatially populations models: current forms and future uses. *Ecological Applications* 5: 3-11.

Goudriaan, J., H.H. Shugart, H. Bugmann, W. Cramer, A. Bondeau, R.A. Gardner, L.A. Hunt, W.K. Lauenroth, J.J. Landsberg, S. Linder, I.R. Noble, W.J. Parton, L.F. Pitelka, M. Stafford Smith, R.W. Sutherst, C. Valentin, and F.I. Woodward. (1999). Use of Models in Global Change Studies (pages 106-140). In: B.H. Walker, W.L. Steffen, J. Canadell and J.S.I. Ingram (eds.). The Terrestrial Biosphere and Global Change: Implications for Natural and Managed Ecosystems. Cambridge University Press, Cambridge.

Hall, C.A.S. and J. Day. (1975). Ecosystem modeling in theory and practice: An introduction with case stories. Second Edition. University Press of Colorado.

Jarvis, P.G. (1995). Scaling processes and problems. *Plant, Cell & Environment* 18: 1079-1089.

Kerr B, M. A. Riley, M. W. Feldman, B.J.M. Bohannan, 2002 "Local dispersal promotes biodiversity in a real-life game of rock-paper-scissors" Nature 418, p171

Murdoch, W.W. (1993). Individual-based models for predicting effects of global change. pp: 147-164. En Biotic Interactions and Global Change., Eds. Kareiva, P. M., J. G. Kingsolver, R.B. Huey, (1993) Sinauer Associates Inc., pp. 559.

Parton, W.J., Stewart, J.W.B. y Cole C.V. (1988). Dynamics of C, N, P and S in grassland soils: a model. *Biochemistry* 5: 109-131.

Peck S.L., (2004). Simulation as experiment: a philosophical reassessment for biological modeling. *Trends in Ecology and Evolution* 19: 530-534.

Schmitz, O. J. (2000). Combining field experiments and individual-based modeling to identify the dynamically relevant organizational scale in a field system. *Oikos* 89:471-484.

Shugart, H.H. and Post W.M. (1992). The potential for application of individual-based simulation models for assessing the effects of global change. En Annual Review on Ecology and Systematics 23: 15-38.

Sinervo B, C.M. Lively 1996 "The rock, paper, scissors game and the evolution of alternative male strategies". Nature 380, p 240

Szabó G. , T. Czárán 2002 "Defensive alliances in spatial models of cyclical population interaction" Phys. Rev. E 64 042902.

Szabó G. and G.A. Sznaider 2004 "Phase transition and selection in a four-species cyclic predator-prey model" Phys. Rev. E 69 031911.

Tainaka K, N Nakagiri 2002 "Segregation in an interacting particle system". Physics Letters A 271, p 92

Vitousek, P.M. (1994). Beyond global warming: ecology and global change. *Ecology* 75: 1861-1876.

Vitousek, P.M., H.A. Mooney, J. Lubchenco and J.M. Melillo. (1997). Human domination of Earth's ecosystems. *Science* 277: 494-499.

Vitousek, P.M., J.D. Aber, R.W. Howarth, G.E. Likens, P.A. Matson, D.W. Schindler, W.H. Schlesinger and D. Tilman. (1997). Human alteration of the global nitrogen cycle: sources and consequences. *Ecological Applications* 7: 737-750.

Vitousek,P.M. (1992). Global environmental change: An Introduction. En Annual Review on Ecology and Systematics 23: 1-14.

Woodward, F. I. (1990). "Global Change: Translating Plant Ecophysiological Responses to Ecosystems." *Trends in Ecology and Evolution* 5(9): 308-311.

Wu, J. and David, J.L. (2002). A spatially explicit hierarchical approach to modeling complex ecological systems: theory and applications. *Ecological Modelling* 153: 7-26.

# Trabajos prácticos

## Bases de datos

El presente trabajo deberá ser presentado y exhibido según cronograma de la guía de TP de la materia, habiéndose designado un día como estado de avance del mismo, también según el mencionado cronograma. El mismo deberá ser funcional, presentar los resultados pedidos y detallar como se llegó a los mismos.

***Presentación de la situación problemática – Opción 1***

Con el objeto de determinar las pautas de manejo sustentable, para aprovechamientos forestales, su consultora es contratada para monitorear los volúmenes anuales aprovechados para cada especie y departamento correspondiente al NEA.

Los informes solicitados son los siguientes (los mismos deberán incluir los gráficos correspondientes):

- Volumen extraído por mes, año, departamento y/o especie.
- Para las principales especies, detallar nivel de extracción según departamento.
- Superficie aprobada en comparación con la superficie pedida por departamento.
- Otros pedidos del profesor durante la presentación del estado de avance.

Cabe destacar que la base de datos deberá contener, para realizar una fiscalización:

- Datos completos de los productores.
- Datos completos de los profesionales que realizaron la presentación de solicitud de desmonte o aprovechamiento.
- Datos completos de los establecimientos donde se realiza el desmonte.
- Otros pedidos del profesor durante la presentación del estado de avance.

***Presentación de la situación problemática – Opción 2***

Con el objeto de evaluar el impacto ambiental de un muelle de transporte de combustibles, una empresa solicita sus servicios de consultoría para evaluar el nivel de contaminación y su influencia sobre la flora y fauna marina.

Los informes solicitados son los siguientes (los mismos deberán incluir los gráficos correspondientes):

- Evolución de poblaciones de especies por mes, año y zona.
- Relación entre nivel de contaminación y cambios en la flora y fauna marina.
- Destacar las zonas de mayor contaminación, para los principales contaminantes.
- Relación entre superficie contaminada y no contaminada a través del tiempo.
- Otros pedidos por el profesor durante la presentación del estado de avance.

Cabe destacar que la base de datos deberá contener como mínimo la siguiente información:

- Datos completos de las diferentes especies involucradas.
- Datos referentes al crecimiento promedio mensual de cada una de las especies.
- Estacionalidad de las especies.
- Capacidad de recuperación frente a disturbios ambientales por contaminantes.
- Concentraciones mínimas máximas aceptables, definidas por organismos oficiales.
- Otros pedidos del profesor durante la presentación del estado de avance.

## Modelos de simulación

El objetivo del ejercicio es que te familiarices con la formulación y uso de un modelo de simulación sencillo que representa un sistema ecológico bajo condiciones de equilibrio y que luego experimenta cambios.

El sistema:

**El propósito del modelo es el de representar la acumulación y tasa de reemplazo de la materia orgánica del suelo.** Esta versión está parametrizada para un pastizal semiárido. El modelo contiene compartimientos de vegetación y de suelo. Los pools o reservorios de materia orgánica del suelo están divididos en dos compartimentos, un pool "activo" y uno "lento", para representar las distintas fracciones de materia orgánica, de acuerdo a su labilidad. Todas las variables de estado en el modelo están expresadas en gramos de materia seca por metro cuadrado, y las tasas en gramos de materia seca por metro cuadrado por año.

Los supuestos:

1. el crecimiento de la biomasa de plantas sigue una función logística (ver al final).
2. la senescencia es la única pérdida de biomasa de las plantas, y es una proporción constante de la biomasa de plantas por año.
3. la descomposición es una proporción constante del tamaño del pool del suelo, y es significativamente mayor en el pool "activo" que en el "lento".

El modelo:

*Variables de estado:*

B= biomasa de plantas
S1= materia orgánica del suelo "activa"
S2= materia orgánica del suelo "lenta"

*Parámetros:*

c= tasa intrínseca de crecimiento de las plantas
s= tasa intrínseca de senescencia
Bmax= máxima cantidad de biomasa de plantas
h= tasa intrínseca de conversión de S1 en humus (S2)
k1= tasa intrínseca de descomposición de S1
k2= tasa intrínseca de descomposición de S2

Tu tarea:

1. A partir de la descripción del modelo hecha más arriba y de la interpretación del modelo implementado en la planilla de Excel adjunta, dibujá un diagrama del sistema, incluyendo todas las variables de estado, de flujo y los controles. Escribí todas las ecuaciones para cada variable de estado.
2. Respondé brevemente las siguientes preguntas
    - ¿Cuáles son las condiciones para que una variable de estado alcance una situación de equilibrio?

- ¿Cuál es el tamaño de las variables de estado (B, S1 y S2) una vez alcanzado el estado de equilibrio? ¿En qué momento (año) ocurre eso, para cada variable de estado?
- Explicá, utilizando una descripción matemática, por qué las variables de estado alcanzan un estado de equilibrio
- ¿Cuáles te parecen que son las limitantes de este modelo respecto a su objetivo?

3. Escribí el modelo en Stella y asegurate que se obtengan los mismos resultados que en la planilla Excel.
4. Implementá el pastoreo (por herbívoros silvestres y/o domésticos) en el modelo, con el objetivo de determinar el efecto del pastoreo sobre la materia orgánica del suelo. Aseguráte de:
    - rehacer el diagrama del sistema
    - explicar los nuevos supuestos
    - formular nuevas ecuaciones y explicar como éstas implementan los supuestos.
    - Poné las nuevas ecuaciones en el modelo (en la planilla Excel o en Stella, como prefieras), hacé gráficos y explicá qué ocurrió y por qué, en relación con el modelo que construiste. ¿Esto realmente representa lo que pensabas que iba a ocurrir como resultado del pastoreo?

Entregá un informe contestando los puntos 1 a 4 en un diskette o por e-mail al docente responsable del turno junto con los archivos de Excel y Stella.

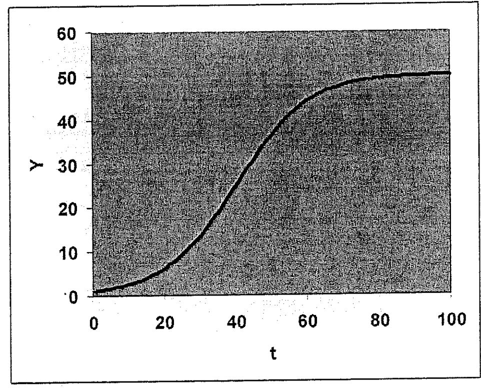
---

El modelo de crecimiento logístico:
Un modelo de crecimiento logístico de *Y* en función del tiempo (*t*) sigue una forma sigmoidea. La tasa de crecimiento de *Y* es máxima en un determinado momento y luego decrece. *Y* se hace asintótica a un valor llamado *Ymax*. La fórmula de cálculo es la siguiente:

$$Y_t = Y_{t-1} + \Delta Y$$

$$\Delta Y = c \cdot Y_{t-1} \cdot \left(1 - \frac{Y_{t-1}}{Y\max}\right)$$

En el caso graficado a la derecha el valor inicial de *Y* es 1, *c* (la tasa intrínseca de crecimiento de Y) es 0.1 y *Ymax* es igual a 50.



# Ejercitación

## 1- Problemas iniciales

1. Construya un modelo conceptual y gráfico que intente resolver el problema de las inundaciones en la ciudad de Buenos Aires.
    a. Qué componentes incluiría en su modelo.
    b. Qué información cree indispensable para abordar este problema.
    c. Qué supuestos establecería.
    d. Cuáles serían los límites espaciales y temporales de su modelo.
    e. Qué relaciones funcionales necesitaría saber para construir su modelo.
    f. Mencioné posibles variables de entrada y de salida de su modelo.

2. Construya un modelo conceptual y gráfico que intente resolver el problema del agotamiento de los recursos pesqueros del mar Argentino. Conteste los mismos ítems que en el problema 1.

## 2- Ejercicio modelo sencillo de red circular

Utilice el modelo de contagio circular (piedra papel y tijera) para contestar las siguientes preguntas

a. ¿Que reglas gobiernan el comportamiento de los agentes?

b. ¿Pueden coexistir las tres poblaciones de individuos? ¿Por cuanto tiempo?

c. ¿Que ocurre con la cantidad de individuos de cada población a lo largo del tiempo?

d. ¿Que ocurre con la distribución espacial de los individuos al aumentar la visibilidad a 5? y a 15?

e. Sobre la base de los puntos anteriores, comente las propiedades emergentes del modelo y explique cómo los parámetros del modelo (visibilidad, número de individuos totales, etc) modifican estas propiedades.

f. ¿Encuentra usted alguna relación entre el modelo utilizado y la "realidad"? En caso afirmativo, especifique cual.

## 3- Ejercicio datos climáticos

***Objetivo:*** Familiarizarse con los datos climáticos que se utilizarán en el primer trabajo práctico de modelado.Dominar herramientas de extracción, filtrado y graficación de datos.

En el archivo clima_NTSG.xls podrán encontrar datos climáticos de distintas localidades. A partir de estos datos:

1) construya una base de datos e ingresen los datos a la misma

2) Calcule los promedios históricos mensuales de Precipitación por localidad. Genere una nueva planilla XLS con esta información

3) Calcule los máximos históricos mensuales de Precipitación por localidad. Genere una nueva planilla XLS con esta información

4) Construya un gráfico que muestre los datos de los puntos 2) y 3) para ~~Coronel Suárez~~ TANDIL

## 4- Ejercicio modelo de balance hídrico

Empleando el modelo de balance hídrico del suelo, conteste las siguientes preguntas:

a. En una pastura de Coronel Suárez, ¿Qué variable presenta mayor variabilidad inter anual, la ETA o la ETP? Realice una figura que fundamente su respuesta. ¿Por qué cree que esto es así?
b. En esta zona de la provincia de Bs. As. suele encontrarse una plancha de tosca sub superficial de carbonato de calcio que limita la exploración del suelo por parte de las raíces. Realice simulaciones para 3 suelos de distinta profundidad (70, 140 y 220 mm de capacidad de almacenaje). ¿Qué puede decir respecto a la Perc (pérdida de agua del sistema) en cada ambiente?
c. ¿Aproximadamente, cuántos días transcurrirían en invierno, y cuántos en verano, para que un suelo con una capacidad de almacenaje de agua útil de 200 mm que se encuentra a capacidad de campo alcance el 40% del % de agua útil, sin que se produzcan precipitaciones en ese lapso?
d. Para un año determinado, realice simulaciones con un incremento del 10% en las temperaturas (máxima y mínima diaria). ¿Qué efecto tiene este cambio sobre la ETP mensual? ¿Y sobre el CAU?
e. Para un mes determinado realice una simulación con una disminución del 10% en la radiación solar incidente. ¿Qué efecto tiene este cambio sobre la ETP? ¿Y sobre el CAU?
f. Discuta los supuestos del modelo. ¿Cuáles le parecen más razonables y cuales cree que pueden comprometer la precisión de los resultados, considerando los objetivos del modelo?

## 5- Ejercicio modelo de dinámica de pasturas (PATUMOD)

Empleando el modelo PATUMOD, conteste las siguientes preguntas:

a. Los disturbios antrópicos modifican generalmente la composición de las comunidades vegetales y pueden alterar las trayectorias susecionales. Qué predice Patumod si se modifican las tasas de extracción de arbustos y árboles.

b. En el mismo contexto del punto anterior, que sucede si cambian el intervalo de extracción de arbustos y árboles para las distintas tasas propuestas previamente.

c. Uno de los parámetros claves para el manejo del pastoreo es la carga animal. Aumentos excesivos de carga en muchos ecosistemas pastoriles han aumentado el componente leñoso de esos sistemas. Qué predice Patumod ante aumentos en este parámetro.

d. ¿Cómo se modifican los resultados si se cambia la altitud?

e. Cómo simularía una práctica agronómica que mejora el valor forrajero del pastizal debajo de los árboles (ej. raleo, fertilización, etc.).

f. busque las relaciones funcionales dentro del modelo que definen el comportamiento de:

    i. Las modificaciones ocasionadas por el pastoreo animal.
    ii. La cobertura de comunidades de mallines.
    iii. La cobertura de comunidades de árboles.
    iv. La transición entre los pastizales y árboles y el barbecho.
    v. La transición entre los barbechos y la comunidad de pastizales.

## 6- Ejercicio contaminación subterránea

Uno de los aspectos ambientales más importantes de la actualidad tiene que ver con la contaminación química de cursos de agua superficiales y/o subterráneos. Distintas actividades industriales y agropecuarias utilizan una diversidad de productos químicos que en menor o mayor medida y en algún punto son vertidos a la red hidrográfica de una cuenca. Dependiendo de una diversidad de factores, este vertido puede resultar inocuo para el medioambiente o contaminar.

La *atrazina* es un herbicida muy utilizado en el cultivo de maíz para el control pre-emergente de malezas. Este agroquímico presenta consecuencias indeseables respecto del medioambiente dada su permanencia en el suelo y lenta degradabilidad (3-6 meses).

Utilice el modelo de dinámica de agua en el suelo visto en clase, para predecir que cantidad de atrazina puede salir del sistema modelado por escorrentía y/o percolación profunda al término de: a) una semana, b) un mes y c) dos meses de la aplicación. Se sabe que originalmente se asperjaron 4 l/ha de formulado (50% PA) el 30 de Septiembre de 2002 en la localidad de Tandil y que la concentración en el agua del suelo es aproximadamente un 70[5]% de la solución original aplicada debido a fenómenos de adsorción por los coloides del suelo. Considere que la tasa de degradación diaria es del 1%

Por favor incluya en el modelo los flujos, variables de estado y controladores necesarios para modelar el comportamiento de la atrazina

[5] Para un tratamiento más profundo y realista sobre este tema puede verse el modelo Macro-DB presentado en "Modelling herbicide treatment impact on groundwater quality in a central Italy area" publicado en 2001 en *Agronomie 21: 267-276* por los autores Businelli, D., Tombesi, E. y Trevisan, M.

## 7- Ejercicio evaluación de modelos

El INTA EEA Montecarlo, la Facultad de Ciencias Forestales y Empresas Forestales de Misiones y Norte de Corrientes desarrollaron un simulador forestal para la provincia de Misiones y el norte de Corrientes que predice diferentes atributos de interés del rodal tales como altura media, diámetro, densidad, área basal y volumen para tres especies comúnmente usadas en la zona: *Pinus elliottii, Pinus taeda y Araucaria angustifolia.* Para correr el modelo es necesario definir la especie, la calidad del sitio y las condiciones iniciales del rodal (como por ejemplo densidad inicial, raleos, etc.).

Una empresa forestal del sur de Corrientes tiene interés en usar dicho modelo para predecir la dinámica de sus rodales y establecer pautas de manejo a largo plazo, pero antes quiere realizar un proceso de validación del modelo ya que el mismo fue desarrollado para una zona cercana pero distinta. Para cumplir con este objetivo se realizaron simulaciones para cuatro diferentes sectores con distintos manejos (Tabla 1) y se compararon con los respectivos valores observados para tres años particulares que se contaba con mediciones a campo (1997, 1999 y 2002; Tabla 2). Además, el Ing. Agr. que asesora dicho establecimiento tiene particular interés en estudiar la sensibilidad del modelo a un parámetro que resulta difícil medir con precisión a campo como es la altura dominante. Para esto realizó varias simulaciones con el modelo para el sector 1 con pequeñas variaciones en este parámetro y registró los resultados simulados (Tabla 3). En base a la información de las tablas 2 y 3, realice los análisis adecuados para validar y evaluar la sensibilidad del modelo al parámetro de interés y decida acerca de la utilidad y alcance del mismo por parte de la empresa forestal.

**Tabla 1.** Características de la implantación y manejo de los distintos sectores.

| *Sectores* | Especie | Año de plantación | Espaciamiento | Raleo |
|---|---|---|---|---|
| 1 | *Pinus elliottii* | 1981 | 2.5 m x 2.5 m | Sin |
| 2 | *Pinus elliottii* | 1982 | 2.5 m x 2.5 m | Sin |
| 3 | *Pinus elliottii* | 1984 | 2.5 m x 2.5 m | Con (33%) |
| 4 | *Pinus elliottii* | 1986 | 2.5 m x 2.5 m | Con (33%) |

**Tabla 2.** Datos observados y simulados por el modelo para distintos sectores y años de 5 variables de respuesta de interés forestal.

| | | Altura media (m) | | Diámetro medio (cm) | | Densidad (arb./ha) | | Área basal ($m^2$/ha) | | Volumen Total c/c ($m^3$/ha) | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Sect. | Año | Obs | Sim | Obs | Sim | Obs | Sim | Obs | Sim | Obs | Sim |
| 1 | 1997 | 17.77 | 18.2 | 20.72 | 19.4 | 1223 | 1223 | 41.13 | 37 | 345.08 | 341 |
| 1 | 1999 | 18.06 | 19.7 | 22.02 | 21 | 1166 | 1151 | 44.24 | 40.6 | 382.64 | 393.1 |
| 1 | 2002 | 20.98 | 21.6 | 23.32 | 23.2 | 1084 | 1049 | 46.08 | 45.1 | 409.51 | 462.4 |
| 2 | 1997 | 18.69 | 17 | 20.25 | 18.8 | 1182 | 1182 | 38.00 | 33.5 | 315.75 | 295.5 |
| 2 | 1999 | 19.07 | 18.6 | 21.56 | 20.4 | 1173 | 1117 | 42.64 | 37.3 | 364.87 | 348.1 |
| 2 | 2002 | 20.70 | 20.6 | 23.24 | 22.7 | 1094 | 1022 | 46.18 | 42 | 409.21 | 418.4 |
| 3 | 1997 | 14.58 | 15.1 | 19.02 | 18.7 | 1384 | 1384 | 39.28 | 39 | 313.81 | 313 |
| 3 | 1999 | 15.05 | 16.9 | 21.32 | 20.3 | 960 | 1302 | 34.26 | 43.2 | 288.87 | 373.9 |
| 3 | 2002 | 19.86 | 19.2 | 22.45 | 22.6 | 1013 | 1181 | 40.12 | 48.2 | 348.16 | 453.3 |
| 4 | 1997 | 14.24 | 13 | 17.54 | 17.4 | 1560 | 1560 | 37.71 | 38 | 290.65 | 272.5 |
| 4 | 1999 | 14.90 | 15 | 19.81 | 19.1 | 1165 | 1470 | 35.89 | 42.9 | 291.70 | 341.7 |
| 4 | 2002 | 18.06 | 17.7 | 21.69 | 21.3 | 1156 | 1330 | 42.71 | 48.6 | 364.70 | 431.6 |

**Tabla 3.** Datos simulados por el modelo para el año 2002 de distintas variables de respuesta de interés forestal variando el parámetro que define la altura dominante del rodal al inicio de la corrida.

| Altura dominante (m) | Altura media (m) | Diámetro medio (cm) | Densidad (arb./ha) | Área basal ($m^2$/ha) | Volumen ($m^3$/ha) |
|---|---|---|---|---|---|
| 20.0 | 21.3 | 23.0 | 1048 | 44.2 | 450.0 |
| 20.2 | 21.6 | 23.2 | 1049 | 45.1 | 462.4 |
| 20.4 | 21.8 | 23.4 | 1049 | 46.1 | 475.1 |
| 20.6 | 22.0 | 23.7 | 1049 | 47.1 | 488.3 |
| 20.8 | 22.3 | 23.9 | 1049 | 48.1 | 501.7 |

## 8- Ejercicio lectura publicación

Consigna y resumen del trabajo "**Modelo orientado a objetos para el manejo de la calidad de agua superficial**" de Elshorbagy Amin y Ormsbee Lindell publicado en Environmental Modelling & Software 2005.

***Resumen***

La falta de datos suficientes acerca de la calidad de agua, en muchos lugares dificulta la modelación de los procesos asociados y por lo tanto afecta la posibilidad del manejo racional del recurso. En este trabajo se presenta y discute el potencial de un ambiente de simulación orientado a objetos para el manejo de la calidad de agua superficial (ríos, arroyos, lagos y lagunas), basado sobre los conceptos de dinámica de sistemas. El modelo, desarrollado en el ambiente Stella©, presenta una estructura simple de reservorio con flujos de entrada y salida. Se considera que la contaminación fecal de una cuenca o sub-cuenca en un punto de la misma depende del ingreso de aguas contaminadas a ese punto, que a su vez depende de la cantidad de sistemas sépticos, el caudal y la ubicación específica dentro de la cuenca (Figura 3). Además, existe un flujo de salida que representa la mortalidad de patógenos en el trayecto del curso de agua. En el trabajo se detallan las principales características de la aproximación empleada y además se presenta un caso de estudio sobre el uso de este modelo para el manejo de la calidad de agua al sudeste de Kentucky, EE.UU., donde se resaltan las características claves de la aproximación. Se presentan los resultados de un procedimiento simple de calibración y validación para el caso de estudio y se emplea el modelo para evaluar cuatro escenarios resultantes de la combinación de diferentes caudales y cantidades de sistemas sépticos. En la última sección del trabajo se discuten las ventajas y desventajas que enfrenta en la actualidad la aproximación con este tipo de modelos para simular el comportamiento de sistemas hidrológicos. Se puntualiza el uso potencial del modelo propuesto, especialmente en condiciones de datos insuficientes, y los desafíos de los hidrólogos para explotar tales aproximaciones de modelado.

***Consigna***

1. Defina claramente el problema que es modelado en este trabajo.
2. ¿A qué se hace referencia cuando se habla de modelo *orientado a objetos?*.
3. Destaque la principal diferencia entre una variable de *estado o reservorio* y una variable de *flujo*.
4. Según la **Figura 3a** del trabajo ¿Qué proceso regula el "*Number of SF*" y de qué atributo real del sistema depende?
5. Según la **Figura 4b** del trabajo ¿qué puede decir acerca de las predicciones efectuadas por el modelo?.
6. ¿Cuál fue el peor escenario simulado en términos de contaminación? (ver **Figura 5**)
7. Según la **Figura 6** ¿Cuál es la sub-cuenca más contaminada?
8. Basándose en el conocimiento que se desprende del trabajo acerca de la hidrología y la contaminación de estos sistemas, ¿Qué variable podría agregar para mejorar las predicciones del modelo?